



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR -TE 141599

**PERANCANGAN KONTROL *FUZZY-LQR* UNTUK
TRACKING ROBOT PENDULUM TERBALIK BERODA
DUA MODEL SEGWAY**

Farid Zainul Hasan
NRP 2214 105 026

Dosen Pembimbing
Ir. Ali Fatoni, MT.
Eka Iskandar, ST, .MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT -TE 141599

***DESIGN OF FUZZY-LQR CONTROL FOR TRACKING
ROBOT ON TWO WHEEL INVERTED PENDULUM
MODELS SEGWAY***

Farid Zainul Hasan
NRP 2214 105 026

Advisor
Ir. Ali Fatoni, MT.
Eka Iskandar, ST.,MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

**PERANCANGAN KONTROL *FUZZY-LQR* UNTUK
TRACKING ROBOT PENDULUM TERBALIK
BERODA DUA MODEL *SEGWAY***

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I



Ir. Ali Fatoni, MT.

NIP. 196206031989031002

Dosen Pembimbing II



Eka Iskandar, ST., MT.

NIP. 198005282008121001



PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Perancangan Kontrol *Fuzzy-LQR* Untuk *Tracking Robot Pendulum Terbalik Beroda Dua Model Segway***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2016

Farid Zainul Hasan
NRP 2214105026

Halaman ini sengaja dikosongkan

PERANCANGAN KONTROL *FUZZY-LQR* UNTUK TRACKING ROBOT PENDULUM TERBALIK BERODA DUA MODEL *SEGWAY*

Nama : Farid Zainul Hasan
Pembimbing : 1. Ir. Ali Fatoni, MT.
2. Eka Iskandar, ST.,MT.

ABSTRAK

Robot pendulum terbalik beroda dua model *segway* merupakan sistem yang tidak stabil dan *non-linier*. Sistem kontrol dibutuhkan untuk menstabilkan dan menjaga robot dapat setimbang di daerah 0 radian. Pada penelitian tugas akhir ini membahas desain kontrol *tracking* robot pendulum terbalik beroda dua model *segway* dengan metode *fuzzy-LQR* kontroler agar sistem dapat bergerak mengikuti sinyal referensi dan dapat mempertahankan posisi robot tegak lurus. Sinyal referensi dalam tahapan simulasi sistem menggunakan sinyal *step*. Model *non-linier* dari sistem robot pendulum terbalik dapat direpresentasikan dengan model *fuzzy* Takagi-Sugeno untuk beberapa kondisi mula, kemudian untuk kontrol *state feedback* disusun menggunakan kontrol optimal. Hasil simulasi menunjukkan bahwa posisi robot pendulum terbalik model *segway* dapat mengikuti sinyal referensi *step*, sedangkan sudut dari robot berada disekitar 0 radian dengan *settling time* 2 detik.

Kata Kunci : *Fuzzy-LQR* kontrol, *Segway*, *Tracking* control.

Halaman ini sengaja dikosongkan

DESIGN OF FUZZY-LQR CONTROLS FOR TRACKING ROBOT ON TWO-WHEEL INVERTED PENDULUM MODELS SEGWAY

Name : Farid Zainul Hasan
Advisor : 1. Ir. Ali Fatoni, MT.
2. Eka Iskandar, ST.,MT.

ABSTRACT

The inverted pendulum robot models two-wheeled is a system that is unstable and nonlinear. The control system is needed to stabilize and maintain the robot in equilibrium at 0 radians region. In this final project discusses the design of inverted pendulum robot control tracking two-wheeled Segway models with fuzzy-LQR controller for the system to move to follow the reference signal and the robot can maintain an upright position. Signal reference in simulation system using a signal step. Non-linear models of inverted pendulum robot system can be represented by Takagi-Sugeno fuzzy model for some conditions first, and then to control state feedback optimal control use prepared. The simulation results showed that the inverted pendulum robot position Segway models can follow the signal reference step, while the angle of the robot was around 0 radians with a settling time of 2 seconds.

Key Word : Fuzzy-LQR control, Segway, Tracking control.

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	iii
HALAMAN PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
1.7 Relevansi	4
BAB 2 TEORI PENUNJANG	5
2.1 Pendahuluan	5
2.2 Balancing Robot	5
2.2.1 Model Fisik Sistem robot	6
2.2.2 Model Matematis	8
2.3 Linearisasi	9
2.4 <i>LQR control</i>	9
2.4.1 Pemilihan Pembobot LQR	9
2.4.2 <i>State Feedback With Reference Trajectory</i>	10
2.4.3 <i>Integral Action Tracking Control</i>	11
2.5 Model <i>Fuzzy</i> Takagi Sudeno	12
2.6 Teori <i>H-Bridge</i>	14
2.6.1 Konfigurasi <i>H-Bridge</i>	14
BAB 3 PERANCANGAN SISTEM	17
3.1 Pembuatan Robot	17
3.1.1 Pembuatan <i>Chassis</i> model segway	17
3.1.2 Pengerjaan komponen pendukung	17

3.2 Linierisasi.....	18
3.3 LQR (<i>linier quadratic regulator</i>).....	19
3.4 <i>Integral Action Tracking Control</i>	20
3.5 Pemodelan <i>Fuzzy</i> Takagi-Sugeno	22
BAB 4 HASIL SIMULASI DAN IMPLEMENTASI.....	25
4.1 Hasil Simulasi	25
4.1.1 Hasil Simulasi Sistem dengan Beberapa Kondisi	
Awal	25
4.1.1.1 Simulasi tanpa menggunakan input	25
4.1.1.2 Respon sudut robot menggunakan input	27
4.1.1.3 Respon posisi robot menggunakan input	28
4.1.1.4 <i>Integral Action Tracking Control</i>	30
4.1.2 Hasil Simulasi Sistem dengan Gangguan Internal Robot	31
4.1.2.1 Respon sudut robot menggunakan input <i>step</i>	32
4.1.2.2 Respon posisi robot menggunakan input	33
4.1.2.3 <i>Integral Action Tracking Control</i>	34
4.2 Implementasi	35
BAB 5 PENUTUP.....	39
5.1 Kesimpulan	39
5.2 Saran	39
DAFTAR PUSTAKA	41
LAMPIRAN 1	43
LAMPIRAN 2	49
LAMPIRAN 3	57
LAMPIRAN 4	59
RIWAYAT HIDUP.....	73

DAFTAR TABEL

Tabel 2.1	Parameter dari robot <i>inverted pendulum</i> model <i>segway</i>	8
------------------	---	---

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1	Robot JOE	5
Gambar 2.2	Robot Nbot dan Robot Legway	6
Gambar 2.3	Gaya-gaya pada robot <i>inverted</i> pendulum model <i>segway</i> .7	
Gambar 2.4	Konsep <i>driver</i> motor <i>H-Bridge</i> MOSFET	12
Gambar 2.5	MOSFET A&D <i>on</i> dan MOSFET B&C <i>off</i>	12
Gambar 2.6	MOSFET A&D <i>off</i> dan MOSFET B&C <i>on</i>	13
Gambar 3.1	Rangkaian Pendukung Elektrik.....	16
Gambar 4.1	Respon sistem tanpa input dengan kondisi mula 0 rad ..	28
Gambar 4.2	Respon sistem tanpa input dengan kondisi mula 0.2 rad	28
Gambar 4.3	Respon sistem sudut dengan input kondisi mula 0 rad	29
Gambar 4.4	Respon sistem sudut dengan input kondisi mula ± 0.2 rad.....	30
Gambar 4.5	Respon sistem posisi dengan input kondisi mula 0 rad	31
Gambar 4.6	Respon sistem posisi dengan input kondisi mula ± 0.2 rad.....	31
Gambar 4.7	Respon <i>tracking</i> posisi dengan input kondisi mula 0 rad	32
Gambar 4.8	Respon <i>tracking</i> posisi dengan input kondisi mula ± 0.2 rad	33
Gambar 4.9	Respon sistem tanpa input dengan kondisi mula 0 rad (<i>disturbance</i>)	34
Gambar 4.10	Respon sistem pada sudut robot menggunakan input dengan kondisi mula 0 rad (<i>disturbance</i>)	35
Gambar 4.11	Respon sistem pada posisi robot menggunakan input dengan kondisi mula 0 rad (<i>disturbance</i>)	36
Gambar 4.12	Respon <i>tracking</i> posisi dengan input kondisi mula 0 rad (<i>disturbance</i>)	37

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi akhir-akhir ini berkembang sangat pesat. Diantaranya ialah teknologi robotika, dimana telah menjangkau sisi hiburan, industri, dan pendidikan. Kerumitan didalam struktur teknologi robotika selalu dipermudah dengan seiring perkembangan zaman, salah satu cara menambah tingkat kecerdasan suatu robot yaitu dengan menambah metode kontrol bahkan memberikan kecerdasan buatan pada robot tersebut. Salah satunya adalah Robot Pendulum Terbalik Beroda Dua Model *Segway*.

Robot Pendulum Terbalik Beroda Dua Model *Segway* merupakan suatu robot *mobile* yang memiliki roda di sisi kanan dan kirinya yang tidak akan seimbang tanpa adanya kontroler. *Tracking* dan stabilisasi robot ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan diatas kendaraan beroda. Proses *tracking* robot memerlukan suatu metode kontrol yang baik untuk mengikuti input sinyal *reference* dan dapat mempertahankan posisi robot dalam keadaan tegak lurus. Bahkan sekarang ini, konsep *tracking* dan stabilisasi robot pendulum terbalik beroda dua telah digunakan sebagai alat transportasi yang bernama *segway*.

Dalam Tugas Akhir ini akan dikaji masalah pengendalian sistem tracking robot pendulum terbalik beroda dua model *segway* dengan menggunakan kontrol *fuzzy-LQR* yang kemudian akan di simulasi-kan ke dalam MATLAB dan di implementasikan.

1.2 Perumusan Masalah

Mempertahankan posisi robot pendulum terbalik beroda dua model *segway* pada sudut sekitar nol radian, dan skema permasalahan yang akan dibahas dalam penelitian Tugas Akhir ialah:

1. Permodelan Robot Pendulum Terbalik Beroda Dua
2. Linierisasi sistem
3. Desain kontroler
4. Simulasi sistem
5. Implementasi sistem

1.3 Batasan Masalah

Masalah yang ditangani dalam Tugas Akhir ini diantaranya meliputi :

1. Desain kontrol *tracking* menggunakan metode *error LQR*
2. Kontroler pendulum terbalik model *segway* dapat mengontrol posisi robot sesuai dengan sinyal referensi yang diberikan.
3. Sinyal referensi yang digunakan berupa sinyal *step*.

1.4 Tujuan

Hasil dari penelitian ini diharapkan mampu membuat desain kontroler menggunakan metode *fuzzy-LQR* yang di terapkan pada proses *tracking* robot pendulum terbalik beroda dua model *segway*. Dengan adanya penerapan kontroler menggunakan *fuzz-LQR* diharapkan dapat meningkatkan ke akuratan dari performansi robot pendulum terbalik beroda dua model *segway*.

1.5 Metodologi

Metodologi penelitian yang digunakan dalam penyelesaian Tugas Akhir ini adalah sebagai berikut:

1. Studi literatur
Mencari dan mempelajari referensi tentang permasalahan *fuzzy*, akses sensor accelerometer MMA7361, dan mencari tinjauan pustaka yang sesuai dengan penelitian tugas akhir yang dilakukan.
2. Konsultasi
Dalam masa perencanaan dan pembuatan sistem dikonsultasi-kan dengan dosen pembimbing, narasumber, dan sumber-sumber lain yang dijadikan acuan.
3. Pemodelan sistem
4. Linierisasi sistem
5. Desain kontroler
6. Simulasi
7. Implementasi & pembahasan
Implementasi sistem meliputi hasil dari rancangan yang telah dibuat, diuji.

1.6 Sistematika Pembahasan

Untuk memudahkan pembahasan, maka Tugas Akhir ini akan dibagi menjadi lima bab dengan sistematika sebagai berikut:

BAB I : PENDAHULUAN

Bab ini meliputi latar belakang, perumusan masalah, batasan masalah, tujuan, sistematika pembahasan, metodologi penelitian, dan relevansi.

BAB II : TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan pustaka pendahuluan, konsep dasar dari sistem inverse pendulum, teknik linierisasi pada model matematika *linear* pada sistem *inverse* pendulum, *fuzzy* T-S, *linear Quadratic Regulator* (LQR).

BAB III : METODE PENELITIAN

Bab ini membahas perancangan simulasi robot inverse pendulum model *segway* dan algoritma kontrol *linear Quadratic Regulator* (LQR) serta perancangan implementasi sistem menggunakan sinyal referensi berupa sinyal *step*.

BAB IV : SIMULASI DAN IMPLEMENTASI SISTEM

Bab ini memuat hasil simulasi serta menganalisanya.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Kontribusi dari penelitian Tugas Akhir ini adalah untuk pengembangan sistem kontrol pada robot pendulum terbalik model *segway* dengan menggunakan *fuzzy* T-S.

Halaman ini sengaja dikosongkan

BAB 2

TEORI PENUNJANG

2.1 Pendahuluan

Perbandingan dari produk *segway* dengan robot model *segway* dalam penelitian kali ini, ialah semua material robot dibentuk dengan meyerupai model *segway*. Sebelum mengkaji literatur untuk pengerjaan tugas penelitian ialah membutuhkan informasi mengenai teknologi dan metodologi sesuai dengan topik. Didalam bagian ini akan disediakan ringkasan dari literatur yang dibutuhkan dengan topik mengenai *tracking* kontrol robot pendulum terbalik model *segway*.

2.2 *Balancing Robot*

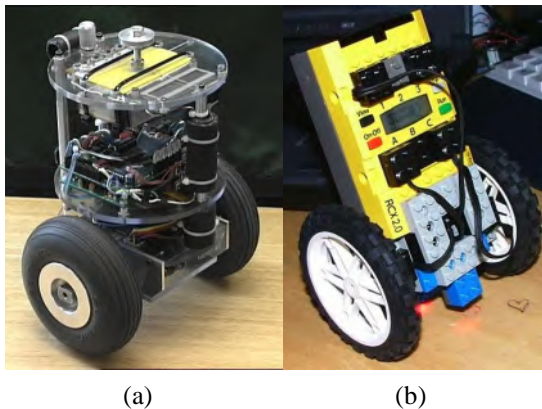
Permasalahan sistem pendulum kereta terbalik sering kita temui di bidang teknik kontrol. Keunikan yang dimiliki dari sistem pendulum kereta terbalik ini, telah menarik minat dari banyak penelitian dan penggemar robotika di seluruh dunia untuk dikembangkan. Dalam beberapa tahun terakhir ini, peneliti mengembangkan aplikasi *inverted pendulum* dengan berbagai permasalahan, diantaranya cara berjalan robot humanoid, robot kursi roda, *segway* (*personal transport systems*).[1,2]

Peneliti di laboratorium elektro industri di *Swiss Federal Institute of Technology* telah membuat prototipe ukuran kecil mengenai *Digital Signal Processor controlled two-wheeled* dasar dari pendulum terbalik dengan beban diberikan untuk mensimulasikan pengendara(manusia). Pengendalian linier state space memanfaatkan informasi sensor dari gyro dan encoder untuk menstabilkan sistem (Grasser et al. 2002).



Gambar 2.1 Robot JOE

Sistem *balancing robot (Inverted pendulum)* memiliki perhatian penting oleh para peneliti dan penggemar robot. Contohnya, Nbot (Gambar 2.2 a) kesetimbangan robot dua roda dibuat serupa dengan JOE, robot ini menggunakan dua informasi sensor ialah posisi dan encoder untuk menjaga kestabilan sistem. Robot Legway (Gambar 2.2 b) dikembangkan oleh Steven Hassenplug dengan memiliki perbedaan konstruksi *balancing robot* dimana hanya menggunakan sensor kemiringan sudut.[1]



Gambar 2.2 Robot Nbot dan Robot Legway

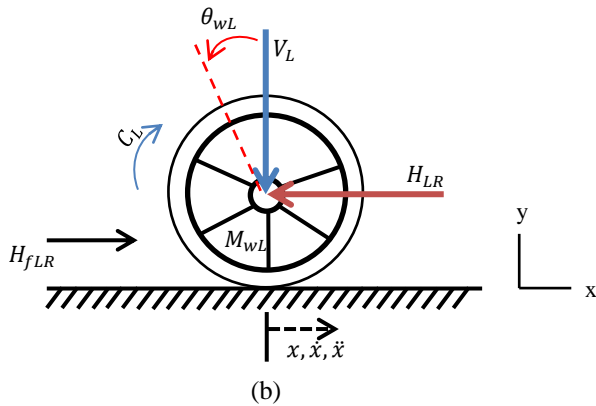
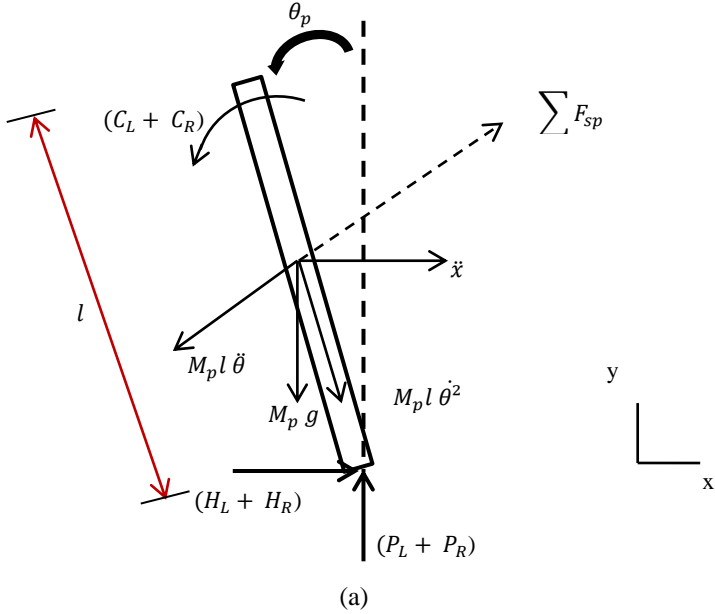
Peningkatan pesat dari populasi usia lanjut (jompo) di Negara seperti jepang telah mendorong para peneliti dalam mengembangkan robot kursi-roda untuk membantu yang lemah untuk berjalan, (Takahashi et al. 2000). Sistem kontrol pada *inverted pendulum* di aplikasikan ketika kursi roda bermanuver pada jalan trotoar dan jalan pendek.

Pada tingkatan yang lebih tinggi, Sugihara et al. (2002) dimana gerakan jalan dari manusia dimodelkan seperti *inverted pendulum* dalam merancang metode baru dalam waktu kontinyu pada robot *humanoid*. [1,2,3]

2.2.1 Model Fisik *Balancing Robot*

Inverted pendulum model segway dan gaya yang bekerja pada *Inverted pendulum* ditunjukkan pada (Gambar 2.3). Skema dari *inverted*

pendulum model segway terdiri dua bagian besar, yaitu robot dan beban untuk mensimulasikan dari (pendulum/manusia). Robot hanya dapat bergerak secara bebas pada area horisontal atau sumbu x.



Gambar 2.3 Gaya-gaya pada robot *inverted pendulum* model segway

Tabel 2.1 Parameter dari robot *inverted pendulum* model *segway*

Nilai	Simbol	Parameter
$g = 9.81$	[m/s ²]	gravitasi bumi
$m_c = 0.51$	[kg]	berat <i>chassis</i>
$m_w = 0.02$	[kg]	berat roda
$j_c = 0.00089424$	[kgm ²]	momen inersia dari <i>chassis</i>
$j_w = 0.00003425$	[kgm ²]	momen inersia dari roda
$R = 0.123$	[m]	jarak antara pusat roda dengan tinggi ujung <i>chassis</i>
$r = 0.0185$	[m]	jari-jari roda
$p = 0.13$	[m]	panjang <i>chassis</i>
$h = 0.121$	[m]	tinggi <i>chassis</i>
$d = 0.08$	[m]	lebar <i>chassis</i>
$l = 0.062$	[m]	jarak pusat gravitasi dengan roda
$\tau_m = 0.6395$	[Nm]	torsi motor kanan dan kiri
$\tau_{m\text{kiri}} = 0.31975$	[Nm]	torsi motor kiri
$\tau_{m\text{kanan}} = 0.31975$	[Nm]	torsi motor kanan

2.2.2 Model Matematis *Balancing Robot*

Sistem robot *inverted pendulum* model *segway* memiliki empat state yaitu x_1 , x_2 , x_3 dan x_4 . Dimana $x_1 = \theta$, ialah sudut robot terhadap jatuhnya robot terhadap bidang horizontal atau sumbu x. $x_2 = \dot{\theta}$, ialah posisi robot. $x_3 = \ddot{\theta}$, ialah kecepatan sudut terhadap jatuhnya robot ke sumbu x atau bidang horizontal. $x_4 = \dot{x}$, ialah kecepatan posisi robot.

Dari hasil model matematika robot *inverted pendulum* model *segway* dapat dibentuk ke persamaan *state*, dibawah ini merupakan persamaan *state* sistem.

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 \dot{x}_3 &= \frac{-M_p l \ddot{x} \cos \theta_p - 2 \tau_m - M_p g l \sin \theta_p}{(J_p + M_p l^2)} \\
 \dot{x}_4 &= \frac{\frac{2 \tau_m}{r} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p}{2M_w + \frac{2J_w}{r^2} + M_p}
 \end{aligned} \tag{2.1}$$

2.3 Linearisasi

Sebelum melanjutkan analisis lebih lanjut, diharuskan untuk me-linearisasikan persamaan state sistem robot *inverted pendulum* model *segway*. Ada dua poin kesetimbangan, $\theta=0$ (pendulum mengarah ke bawah, stabil) dan $\theta=\pi$ (pendulum mengarah ke atas, tidak stabil). Dalam perancangan untuk linierisasi model sistem menggunakan metode deret Taylor. [4]

2.4 LQR Control

Teori mengenai kontrol *linier* kuadratik didalam plant yang akan saya rancang, digunakan untuk mencari gain *state feedback* dan permasalahan pada *state feedback with reference trajectory*.

2.4.1 Pemilihan Pembobot LQR

$$\dot{x} = Ax + Bu \quad J = \int_0^\infty \underbrace{(x^T Q x + u^T R u + x^T S u)}_{L(x,u)} dt \quad (2.2)$$

Pendekatan pemilihan pembobot Q dan R:

1. Pemilihan secara sederhana: $Q = I$, $R = \rho I \rightarrow L = \|x\|^2 + \rho \|u\|^2$. Pengubahan nilai ρ untuk membuat nilai respon lebih baik.
2. Pembobot diagonal

$$Q = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix} \quad R = \rho \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_n \end{bmatrix}$$

Pemilihan masing-masing bobot Q diberikan sama dengan nilai keburukan / error di keadaan yang mempengaruhi sistem. x_1 = sudut dalam radian, x_2 = jarak dalam meter.

$$\frac{1}{60} \text{ rad error} \rightarrow q_1 = (60)^2 \quad \text{dimana } q_1 x_1^2 = 1 \text{ ketika } x_1 = \frac{1}{60}$$

$$1 \text{ cm error} \rightarrow q_2 = \left(\frac{1}{100}\right)^2 \quad \text{dimana } q_2 x_2^2 = 1 \text{ ketika } x_2 = 1 \text{ cm}$$

Samahalnya dengan R, gunakan ρ untuk memberikan input / kesetimbangan *state*.

3. Metode *trial and error* (coba-coba).

2.4.2 *State Feedback With Reference Trajectory*

Dimana didalam suatu sistem $\dot{x} = f(x,u)$ dan trayektori yang diinginkan (x_d, u_d) . Diinginkan desain kompensator dari persamaan $u = \alpha(x, x_d, u_d)$ dengan $\lim_{t \rightarrow \infty} x - x_d = 0$. diketahui bahwa ini merupakan *trajectory tracking problem*.

Untuk desain kontroler sendiri, dikonstruksi dari sistem *error*. Dengan mengasumsikan $f(x,u) = f(x) + g(x)u$ (*state* sistem merupakan nonlinier, tetapi linier di dalam input; kasus ini sering diaplikasikan).

$e = x - x_d$, $v = u - u_d$ dan komputasi dimanis untuk error.

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{x}_d = f(x) + g(x)u - f(x_d) + g(x_d)u_d \\ &= f(e+x_d) - f(x_d) + g(e+x_d)(v+u_d) - g(x_d)u_d \\ &= F(e, v, x_d(t), u_d(t))\end{aligned}\quad (2.3)$$

Umumnya, sistem ini adalah *time varying*.

Utuk *trajectory tracking*, diasumsikan bahwa nilai e kecil (jika kontroler bekerja dengan baik) dan dapat dilinierisasikan sekitar $e=0$;

$$\dot{e} \approx A(t)e + B(t)v \quad (2.4)$$

dimana:

$$A(t) = \frac{\partial F}{\partial e} \Big|_{(x_d(t), u_d(t))} \quad B(t) = \frac{\partial F}{\partial v} \Big|_{(x_d(t), u_d(t))}$$

Sering mengenai kasus *tracking* bahwa $A(t)$ dan $B(t)$ hanya mengandalkan dari *state* x_d , dimana dalam kasus ini cocok di tulis $A(t) = A(x_d)$ dan $B(t) = B(x_d)$.

Asumsi sekarang nilai x_d dan u_d masing-masing adalah konstan atau sembarang (respek terhadap kreteria *performancy*). Diiijinkan dalam mempertimbangan (nilai konstan) sistem linier diberikan dari

$A(x_d)$ dan $B(x_d)$. Jika desain dari kontrol umpan balik $K(x_d)$, jadi dapat digunakan sistem menggunakan umpan balik

$$v = K(x_d) e \quad (2.5)$$

Substitusi kembali dari definisi e persamaan (2.4) dan v persamaan (2.5), kontroler akan menjadi

$$u = K(x_d) (x - x_d) + u_d \quad (2.6)$$

persamaan diatas dapat dikatakan sebagai *gain scheduled* kontrol linier dengan *feedforward* u_d .

Dalam kasus sepesial dari sistem linier

$$\dot{x} = Ax + Bu$$

Terlihat mudah melihat bahwa identik dinamis error dari sistem dinamis ($\dot{e} \approx A(t)e + B(t)v$). Didalam kasus robot *inverted pendulum* model *segway* tidak membutuhkan dasar *gain schedule* x_d ; dapat disederhanakan perhitungan gain K konstan dan dapat ditulis persamaannya ialah:

$$u = K (x - x_d) + u_d \quad (2.7)$$

2.4.3 Integral Action Tracking Control

Pendekatan dasar di dalam umpan balik integral dibentuk dari *state* tidak lebih dari kontroler dengan perhitungan integral dari sinyal *error*, yang mana menggunakan teori dari umpan balik.

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix} \quad (2.8)$$

state z merupakan integral *error* dari output diinginkan (r), dan output aktual (y). Catatan didalam penggunaan metode ini jika mendapati *compensator* proses menstabilkan sistem seharusnya akan didapat $\dot{z} = 0$ dalam keadaan *steady state* dan $y = r$ diberikan *augmented* sistem, desain *state space* kontroler dengan cara menggunakan hukum control:

$$u = -K(x - x_d) - K_i z + u_d \quad (2.9)$$

akhir dari kompensator:

$$u = -K(x - x_d) - K_i z + u_d \quad (2.10)$$

$$\dot{z} = y - r \quad (2.11)$$

2.5 Model Fuzzy Takagi-Sugeno [5,6]

Dinamika model fuzzy Takagi-Sugeno (T-S) dapat mempresentasikan hubungan linear input dan output dari sistem yang *nonlinear*. Model fuzzy (T-S) merupakan model *fuzzy* yang sering dipakai dalam proses inferensi *fuzzy*. Dalam banyak hal, *fuzzy* (T-S) hampir sama dengan *fuzzy* tipe mamdani.

Proses fuzzifikasi, operasi logika *fuzzy*, dan implikasinya sama dengan yang ada pada *fuzzy* tipe mamdani. Perbedaan antara kedua *fuzzy* tersebut terletak pada jenis keanggotaannya dalam bagian konsekuen. Model fuzzy (T-S) menggunakan fungsi keanggotaan output bersifat linear atau konstan. Pada model *fuzzy* T-S, bagian premis dari aturan **If-Then** merupakan aturan model *plant* yang dapat ditulis dalam bentuk sebagai berikut:

aturan *plant* ke- i :

If $z_1(t)$ adalah M_{i1} **and ... and** $z_j(t)$ adalah M_{ij}

Then $\dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i u(t)$

$y(t) = \mathbf{C}_i \mathbf{x}(t)$

$i = 1, 2, \dots, r \quad j = 1, 2, \dots, p$

dengan r sebagai jumlah aturan *fuzzy*, dan p adalah jumlah himpunan *fuzzy* dalam satu aturan, dan F_{ij} sebagai himpunan *fuzzy*, dengan vektor *state* $\mathbf{x}(t) \in R^n$, vektor kontrol masukan $u(t) \in R^m$, dan vektor keluaran sistem $y(t) \in R^q$, sedangkan $z(t) \in R^j$ merupakan variabel pada bagian premis yang dapat berupa fungsi dari variabel *state*, dan gangguan eksternal. Setelah defuzzifikasi, maka model *fuzzy* T-S secara keseluruhan dapat dituliskan sebagai berikut:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r w_i(z(t)) [\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i u(t)] \quad (2.12)$$

$$y(t) = \sum_{i=1}^r w_i(z(t)) [\mathbf{C}_i \mathbf{x}(t)] \quad (2.13)$$

dengan

$$w_i(z(t)) = \frac{\mu_i(z(t))}{\sum_{i=1}^r \mu_i(z(t))}; \mu_i(z(t)) = \prod_{j=1}^p M_{ij}(z_j(t)) \quad (2.14)$$

Pembobot $w_i(z(t))$ dan derajat keanggotaan $\mu_i(z(t))$ memiliki sifat sebagai berikut :

$$w_i(z(t)) \geq 0 ; \sum_{i=1}^r w_i(z(t)) = 1 \quad (2.15)$$

$$\mu_i(z(t)) \geq 0 ; \sum_{i=1}^r \mu_i(z(t)) > 0 \quad (2.16)$$

Pada sistem ini, kontroler *state feedback* dapat disusun dengan konsep *Parallel Distributed Compensation* (PDC). Dalam konsep PDC, tiap aturan kontroler dirancang berdasarkan aturan model *plant* linear yang bersesuaian dengan himpunan *fuzzy* yang sama. Untuk setiap aturan, dapat digunakan teknik kontrol linear. Hasil keseluruhan kontroler *fuzzy* yang merupakan nonlinear secara umum menjadi perpaduan dari hasil setiap kontroler linear. Dari aturan *plant* yang ada, dapat disusun aturan kontroler *fuzzy* dengan konsep PDC sebagai berikut:

aturan kontroler ke- i :

If $z_1(t)$ is M_{i1} **and** ... **and** $z_j(t)$ is M_{ij}

Then $u(t) = -\mathbf{K}_i \mathbf{x}(t)$

$$i = 1, 2, \dots, r \quad j = 1, 2, \dots, p$$

Secara keseluruhan, keluaran dari kontroler *fuzzy* dapat dituliskan sebagai berikut:

$$u(t) = \sum_{i=1}^r w_i(z(t))[-\mathbf{K}_i \mathbf{x}(t)] \quad (2.17)$$

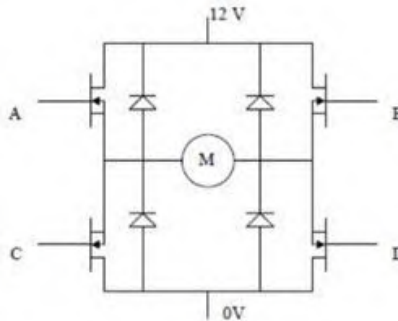
Maka, model *fuzzy* T-S dapat dituliskan menjadi sistem lump tertutup sebagai berikut:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r \sum_{j=1}^p w_i(z(t)) w_j(z(t)) [(\mathbf{A}_i - \mathbf{B}_i \mathbf{K}_j) \mathbf{x}(t)] \quad (2.18)$$

$$y(t) = \sum_{i=1}^r \sum_{j=1}^p w_i(z(t)) w_j(z(t)) [(\mathbf{C}_i - \mathbf{D}_i \mathbf{K}_j) \mathbf{x}(t)] \quad (2.19)$$

2.6 Teori *H-Bridge* MOSFET

H-bridge adalah sebuah perangkat keras berupa rangkaian yang berfungsi untuk menggerakkan motor. Rangkaian ini diberi nama *H-bridge* karena bentuk rangkaiannya yang menyerupai huruf H seperti pada Gambar 2.4.

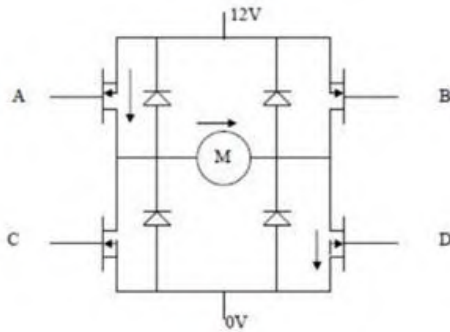


Gambar 2.4 Konsep Driver Motor *H-Bridge* MOSFET

2.6.1 Konfigurasi *H-Bridge* MOSFET

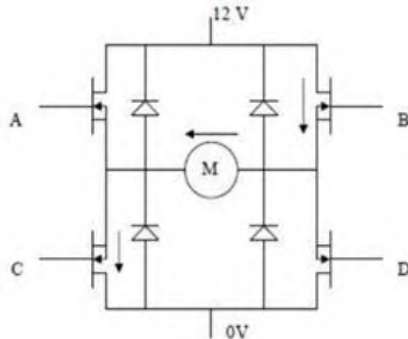
Rangkaian ini terdiri dari dua buah MOSFET kanal P dan dua buah MOSFET kanal N. Prinsip kerja rangkaian ini adalah dengan mengatur mati-hidupnya ke empat MOSFET tersebut. Huruf M pada gambar adalah motor DC yang akan dikendalikan. Bagian atas rangkaian akan dihubungkan dengan sumber daya kutub positif, sedangkan bagian bawah rangkaian akan dihubungkan dengan sumber daya kutub negatif.

Pada saat MOSFET A dan MOSFET D *on* sedangkan MOSFET B dan MOSFET C *off*, maka sisi kiri dari gambar motor akan terhubung dengan kutub positif dari catu daya, sedangkan sisi sebelah kanan motor akan terhubung dengan kutub negatif dari catu daya sehingga motor akan bergerak searah jarum jam dijelaskan pada Gambar 2.5.



Gambar 2.5 MOSFET A & D *on* dan MOSFET B & C *off*

Sebaliknya, jika MOSFET B dan MOSFET C *on* sedangkan MOSFET A dan MOSFET D *off*, maka sisi kanan motor akan terhubung dengan kutub positif dari catu daya sedangkan sisi kiri motor akan terhubung dengan kutub negatif dari catu daya. Maka motor akan bergerak berlawanan arah jarum jam dijelaskan pada Gambar 2.6.



Gambar 2.6 MOSFET A & D *off* dan MOSFET B & C *on*

Halaman ini sengaja dikosongkan

BAB 3

PERANCANGAN SISTEM

Perancangan sistem robot pendulum terbalik beroda dua model *segway* dimulai dari pembuatan robot model *segway* dengan tujuan mendapatkan parameter sistem untuk digunakan dalam proses permodelan sistem. Dari hasil yang didapat pada permodelan sistem dinamis plant, maka perlunya melinierisasikan model matematis dari plant sistem dengan *metode taylor*. Linierisasi dilakukan didaerah kerja π rad (pendulum *up*).

Kontrol optimal digunakan untuk mendapatkan nilai *gain feedback* kontroler. Setelah itu, dijelaskan prosedur kontrol optimal terhadap permasalahan *trajectory* dengan menggunakan konsep *Integral Action*. Kontroler *fuzzy* digunakan untuk tuning nilai *gain feedback* kontroler pada operasi kontrol optimal.

3.1 Pembuatan Robot

Tahapan-tahapan dalam pembuatan robot pendulum terbalik beroda dua model *segway* terbagi menjadi dua langkah pengerjaan, ialah pembuatan *chassis* robot dan pengerjaan kebutuhan elektronik pendukung robot (driver motor, mikrokontroler, modul sensor Accelero MMA7361).

3.1.1 Pembuatan Chassis Robot Invers Pendulum Model Segway

Didalam pembuatan *chassis* robot ini saya menggunakan bahan akrilik sebagai bahan dasar bodi robot. Dimana didalam pembuatan bodi robot kami memperhitungkan panjang, lebar, dan tinggi dari bodi. Langkah – langkah dalam pembuatan *chassis* robot saya memberikan bahan lampiran dibelakang dari buku ini. [Lampiran 1]

3.1.2 Pengerjaan Komponen Pendukung Robot

Pembuatan komponen pendukung elektrik dari robot pendulum terbalik beroda dua model *segway* ialah: rangkaian *low pass filter*, *power supply*, modul sensor accelero MMA7361 dan modul mikro-prosesor Arduino.

Diperoleh persamaan berdasarkan persamaan (25) dan persamaan (27) pada [Appendix 2]

$$\dot{x}_1 = x_3 \quad (3.3)$$

$$\dot{x}_2 = x_4 \quad (3.4)$$

$$\dot{x}_3 = -\frac{M_p g l \theta}{\beta - (\frac{M_p^2 l^2}{\alpha})} + \frac{\frac{2 M_p l}{r \alpha} - 2}{\beta - (\frac{M_p^2 l^2}{\alpha})} \tau_m \quad (3.5)$$

$$\dot{x}_4 = \frac{(M_p^2 g l^2) \theta}{\beta \alpha - M_p^2 l^2} + \frac{2\beta - 2 M_p l r}{r (\beta \alpha - M_p^2 l^2)} \tau_m \quad (3.6)$$

Persamaan *state variable*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{M_p g l \theta}{\beta - (\frac{M_p^2 l^2}{\alpha})} & 0 & 0 & 0 \\ \frac{(M_p^2 g l^2) \theta}{\beta \alpha - M_p^2 l^2} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\frac{2 M_p l}{r \alpha} - 2}{\beta - (\frac{M_p^2 l^2}{\alpha})} \\ \frac{2\beta - 2 M_p l r}{r (\beta \alpha - M_p^2 l^2)} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -203.8267 & 0 & 0 & 0 \\ 8.5917 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 370.7894 \\ 214.9381 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

3.3 Linier Quadratic Optimal (LQR)

Linier kuadratik optimal merupakan teknik *state space* modern untuk desain regulator optimal. LQR mengenai sistem linier dan index performansi berdasarkan:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad (3.7)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (3.8)$$

Performansi dari sistem LQR dapat mewakili dari index performansi integral (persamaan 3.9).

$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t)) dt \quad (3.9)$$

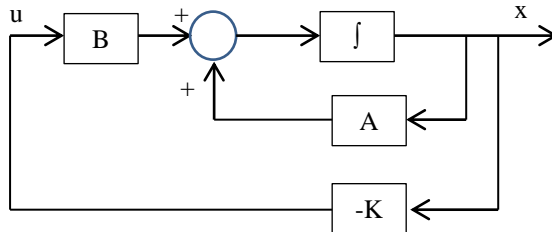
Hukum kontrol untuk LQR menetapkan :

$$u = -R^{-1}B^T\bar{P}x \quad (3.10)$$

dimana $\bar{P} = \bar{P}^T \geq 0$ diselesaikan dengan persamaan aljabar ricatti.

$$0 = PA + A^TP - PBR^{-1}B^TP + Q \quad (3.11)$$

Vector gain $K = -R^{-1}B^T\bar{P}$ didapat nilai dari kontrol umpan balik sistem. Blok diagram dari LQR kontroler seperti di Gambar



Gambar 3.2 Blok Diagram LQR kontrol

Menggunakan MATLAB, untuk persamaan ricatti didapatkan hasil dan gain K kontroler di evaluasi dengan dengan mengubah nilai pembobot Q dan R (*trial and error* / coba-coba).

3.4 Integral Action Tracking Control

Pendekatan dasar di dalam umpan balik integral dibentuk dari *state* tidak lebih dari kontroler dengan perhitungan integral dari sinyal *error*, yang mana menggunakan teori dari umpan balik.

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix} \quad (3.12)$$

state z merupakan integral *error* dari output diinginkan (r), dan output aktual (y). Catatan didalam penggunaan metode ini jika mendapati *compensator* proses menstabilkan sistem seharusnya akan didapat $\dot{z} = 0$ dalam keadaan *steady state* dan $y = r$

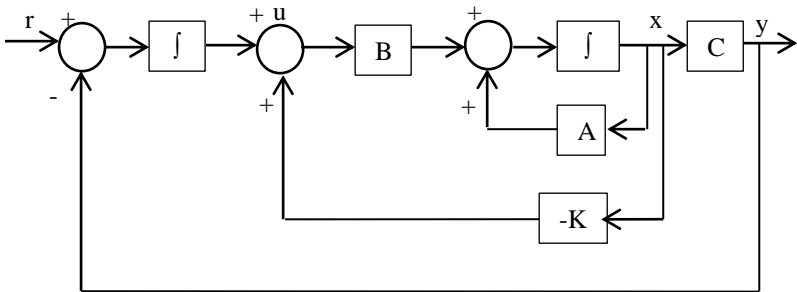
$$A_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 203.8267 & 0 & 0 & 0 \\ 8.5917 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 \\ 370.7894 \\ 214.9381 \\ 0 \end{bmatrix}$$

$$C_i = [0 \quad 0 \quad 1 \quad 0]$$

$$D_i = [0]$$

$$X_{id} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \longrightarrow \text{state yang diinginkan}$$



Gambar 3.2 Blok Diagram LQR integral action kontrol

3.5 Pemodelan *Fuzzy* Takagi-Sugeno

Dari hasil linearisasi SPK, dapat disusun model *Fuzzy* T-S pada saat kondisi mula. Dengan melinearisasi pada dua titik kerja, yaitu $x_2^*=0$ rad dan $x_2^*=\pm 0,2$ rad, maka model *Fuzzy* T-S yang digunakan memiliki tiga aturan dengan satu variabel premis, yaitu sudut pendulum. Dengan menggunakan model linear, maka model *Fuzzy* T-S dibentuk dengan aturan sebagai berikut:

Aturan *plant* ke-1:

If $x_2 = M_1$ (sekitar 0 rad)

Then $\dot{\mathbf{x}}(t) = \bar{\mathbf{A}}_1 \bar{\mathbf{x}}(t) + \bar{\mathbf{B}}_1 u(t)$

$y(t) = \mathbf{C}_1 \mathbf{x}(t)$

Aturan *plant* ke-2:

If $x_2 = M_2$ (sekitar $\pm 0,2$ rad)

Then $\dot{\mathbf{x}}(t) = \bar{\mathbf{A}}_2 \bar{\mathbf{x}}(t) + \bar{\mathbf{B}}_2 u(t)$

$y(t) = \mathbf{C}_2 \mathbf{x}(t)$

Dengan menggunakan konsep PDC, dapat disusun aturan kontroler *Fuzzy* yang berseuaian dengan aturan *plant* menjadi sebagai berikut:

Aturan kontroler ke-1:

If $x_2 = M_1$ (sekitar 0 rad)

Then $u_1 = -\mathbf{K}_1 \bar{\mathbf{x}}(t)$

Aturan kontroler ke-2:

If $x_2 = M_2$ (sekitar $\pm 0,2$ rad)

Then $u_2 = -\mathbf{K}_2 \bar{\mathbf{x}}(t)$

Fungsi keanggotaan yang dipilih adalah fungsi keanggotaan segitiga. M_1 dan M_2 adalah fungsi keanggotaan segitiga dengan parameter fungsi keanggotaan segitiga $\{a;b;c\}=\{-0,2;0;0,2\}$.

Dalam Tugas Akhir ini digunakan inferensi *Fuzzy* dengan penghubung *AND* dan metode defuzifikasi yang digunakan adalah rerata berbobot. Bentuk model *Fuzzy* T-S secara keseluruhan sebagai berikut:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^2 w_i(x_2) [\bar{\mathbf{A}}_i \bar{\mathbf{x}}(t) + \bar{\mathbf{B}}_i u(t)] \quad (3.13)$$

$$y(t) = \sum_{i=1}^2 w_i(x_2) \mathbf{C}_i \mathbf{x}(t) \quad (3.14)$$

Keluaran dari kontroler *Fuzzy* dituliskan sebagai berikut:

$$u(t) = \sum_{i=1}^2 w_i(x_2)[-K_i \bar{x}(t)] \quad (3.15)$$

dengan

$$w_i(x_2(t)) = \frac{\mu_i(x_2(t))}{\sum_{i=1}^3 \mu_i(x_2(t))}; \mu_i(x_2) = M_i(x_2(t)) \quad (3.16)$$

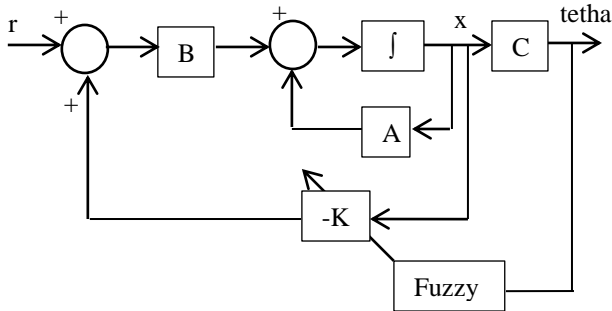
Substitusi Persamaan (3.13) ke (3.15) akan menghasilkan:

$$\begin{aligned} \dot{\bar{x}}(t) &= \sum_{i=1}^2 \sum_{j=1}^2 w_i(x_2(t)) w_j(x_2(t)) [(\bar{A}_i - \bar{B}_i K_j) \bar{x}(t)] \\ y(t) &= \sum_{i=1}^2 w_i(x_2(t)) [C_i \bar{x}(t)] \end{aligned}$$

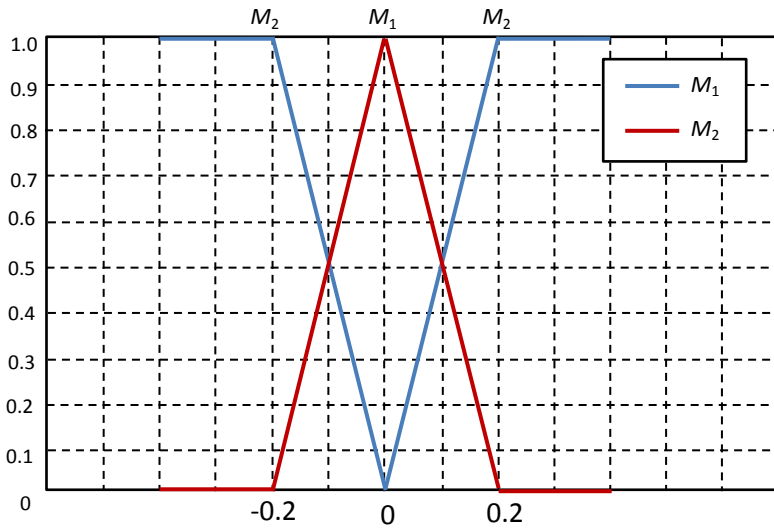
Sinyal kontrol yang dihasilkan dari model *Fuzzy* T-S dapat dituliskan menjadi sebagai berikut:

$$u(t) = M_1(x_2(t))[-K_1 \bar{x}(t)] + M_2(x_2(t))[-K_2 \bar{x}(t)] \quad (3.17)$$

$$u(t) = \sum_{i=1}^2 M_i(x_2)[-K_i \bar{x}(t)] \quad (3.18)$$



Gambar 3.2 Blok Diagram Tuning gain K dengan Fuzzy



Gambar 3.1 Fungsi Keanggotaan M_1 dan M_2

➤ Untuk kondisi mula 0 rad

$$K = [165.9369 \quad -105.3565 \quad 11.7108 \quad -17.9079] \text{ (kondisi mula 0 rad)}$$

$$K_i = [677.6583 \quad 49.2978 \quad -42.8869 \quad -3.1623]$$

➤ Untuk kondisi mula 0.2 rad

$$K = [184.6081 \quad -105.3565 \quad 13.9866 \quad -18.6792] \text{ (kondisi mula 0.2 rad)}$$

$$K_i = [772.1329 \quad 59.9641 \quad -44.7633 \quad -3.1623]$$

BAB 4

PENGUJIAN SIMULASI DAN IMPLEMENTASI

Pada bagian ini akan diterangkan mengenai hasil simulasi dan implementasi sistem beserta analisisnya. Konsep keseluruhan dari robot pendulum terbalik beroda dua model *segway* telah dibahas dalam perancangan sistem pada Bab III dan hasil implementasinya pada *plant* nyata, robot pendulum terbalik beroda dua model *segway* yang saya lampirkan dibagian [Lampiran 1].

4.1 Hasil Simulasi

Dalam mensimulasikan model *plant*, saya menggunakan perangkat lunak MATLAB R2009b. Model gaya gesek pada simulasi diabaikan. Kondisi awal posisi kereta diberikan nilai sebesar 0 meter dan posisi pendulum 0 radian. Sinyal referensi yang diberikan adalah sinyal *step*. Setelah melakukan beberapa percobaan dalam mencari nilai **Q** dan **R**, maka didapat pembobot **Q** dan **R** yang optimal ialah untuk kondisi mula 0 rad dan kondisi mula 0.2 rad. [Lampiran 4]

4.1.1 Hasil Simulasi Sistem dengan Beberapa Kondisi Awal

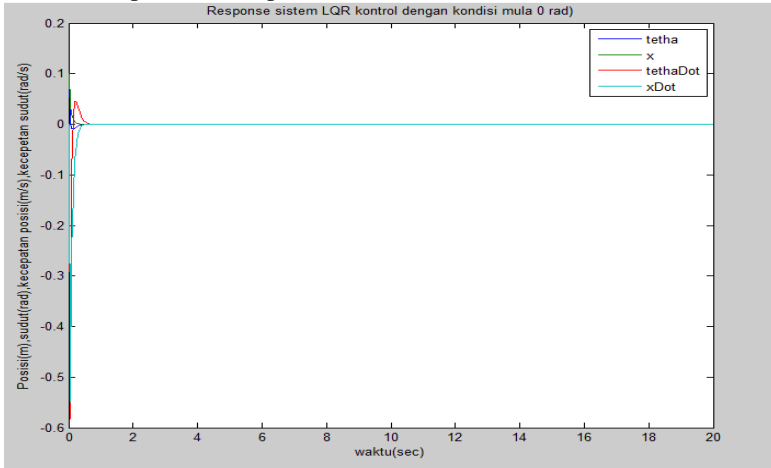
Untuk mendapatkan nilai **Q** dan **R** yang sesuai harus dilakukan percobaan satu-persatu (*trial and error*) dengan menggunakan nilai **Q** yang berbeda-beda baik secara simulasi dengan nilai **R** yang tetap yaitu satu. Performansi respon sudut pendulum dikatakan baik apabila waktu yang dibutuhkan untuk mencapai posisi 0 rad semakin cepat disertai nilai *error* sekecil mungkin. Simulasi dan implementasi kontrol *tracking* pada robot pendulum terbalik beroda dua model *segway* dikatakan baik apabila Respon posisi robot mampu mengikuti sinyal referensi yang diberikan serta mampu mempertahankan posisi pendulum pendulum pada sudut 0 rad.

Simulasi pertama yang dilakukan dalam Tugas Akhir ini menggunakan kondisi awal sudut. Sinyal referensi yang digunakan meliputi sinyal *step*.

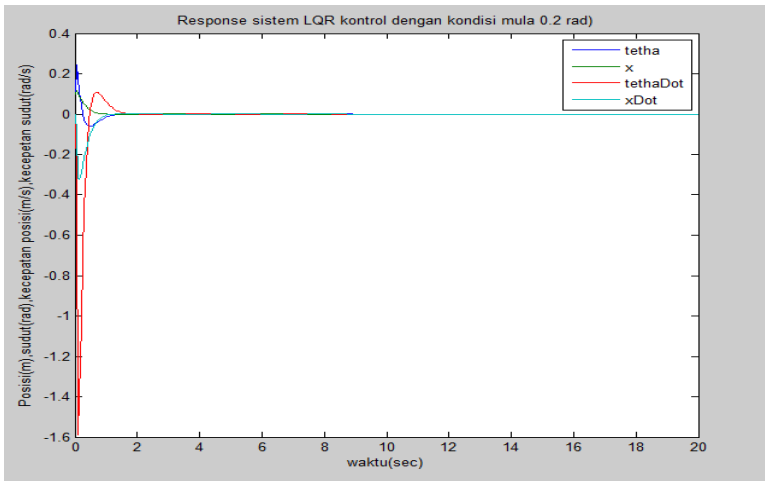
4.1.1.1 Simulasi tanpa menggunakan input *reference*

Kondisi mula ini tanpa menggunakan input dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]^T$), beban robot (0,51 kg) dan panjang robot (0,062 m). Untuk kondisi mula 0 rad nilai pembobot **Q** dan **R** optimal yang

digunakan adalah: $(Q = 1800, 11100, 1, 1)$ dan $(R = 1)$. Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: $(Q = 700, 22500, 1, 1)$ dan $(R = 1)$. Hasil simulasi untuk kondisi mula 0 rad dan ± 0.2 rad dapat dilihat seperti Gambar 4.1 dan Gambar 4.2:



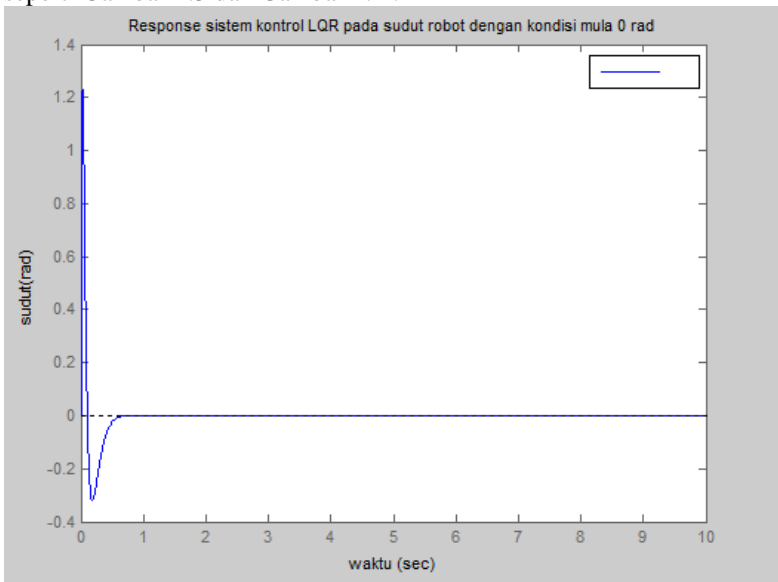
Gambar 4.1 Respon sistem tanpa input dengan kondisi mula 0 rad



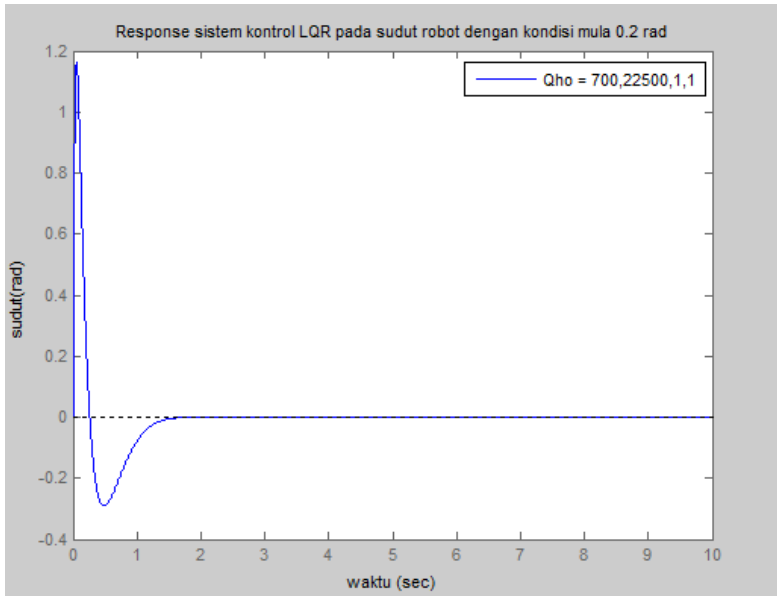
Gambar 4.2 Respon sistem tanpa input dengan kondisi mula 0.2 rad

4.1.1.2 *Response sudut robot menggunakan input step*

Kondisi mula terhadap sudut ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg), panjang robot (0,062 m) dan *state* diinginkan ($x_d = [1 \ 0 \ 0 \ 0]$). Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 1800, 11100, 1, 1) dan (R = 1). Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: (Q = 700, 22500, 1, 1) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dan ± 0.2 rad dapat dilihat seperti Gambar 4.3 dan Gambar 4.4 :



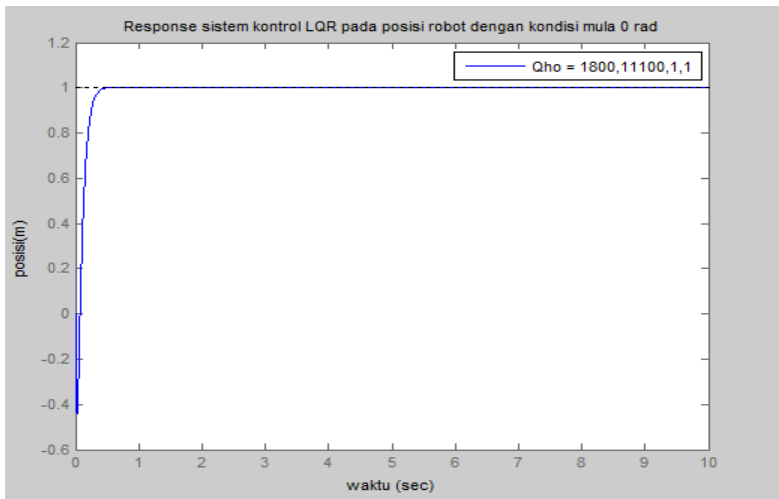
Gambar 4.3 Respon sistem pada sudut robot menggunakan input dengan kondisi mula 0 rad



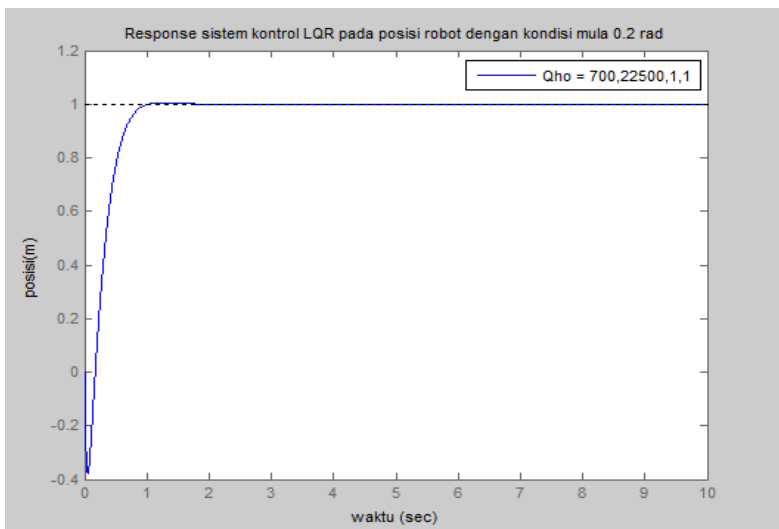
Gambar 4.4 Respon sistem pada sudut robot menggunakan input dengan kondisi mula ± 0.2 rad

4.1.1.3 *Response posisi robot menggunakan input step*

Kondisi mula terhadap posisi ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg), panjang robot (0,062 m), dan *state* output diinginkan ($y_d = [0 \ 1 \ 0 \ 0]$). Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 1800, 11100, 1, 1) dan (R = 1). Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: (Q = 700, 22500, 1, 1) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dan ± 0.2 rad dapat dilihat seperti Gambar 4.5 dan Gambar 4.6 :



Gambar 4.5 Respon sistem pada posisi robot menggunakan input dengan kondisi mula 0 rad



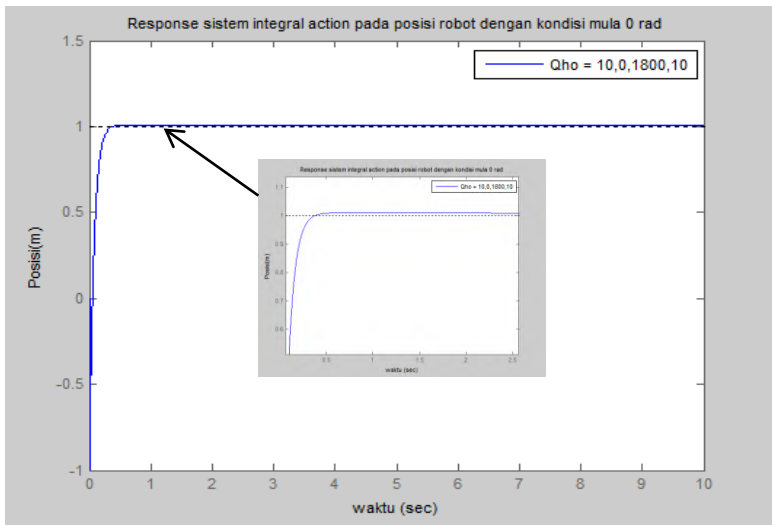
Gambar 4.6 Respon sistem pada posisi robot menggunakan input dengan kondisi mula ± 0.2 rad

4.1.1.4 Integral Action Tracking Control

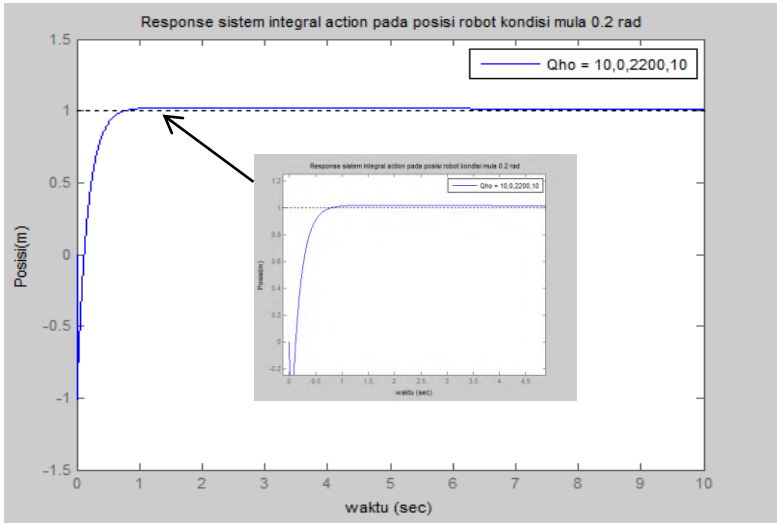
Pendekatan dasar di dalam umpan balik integral dibentuk dari *state* tidak lebih dari kontroler dengan perhitungan integral dari sinyal *error*, yang mana menggunakan teori dari umpan balik.

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix}$$

Kondisi mula terhadap posisi *tracking* ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg), panjang robot (0,062 m), dan *state* diinginkan $x_{id} = [0; 0; 1; 0]$. Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 10, 0, 1800, 10) dan (R = 1). Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: (Q = 10, 0, 2200, 10) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dan ± 0.2 rad dapat dilihat seperti Gambar 4.7 dan Gambar 4.8:



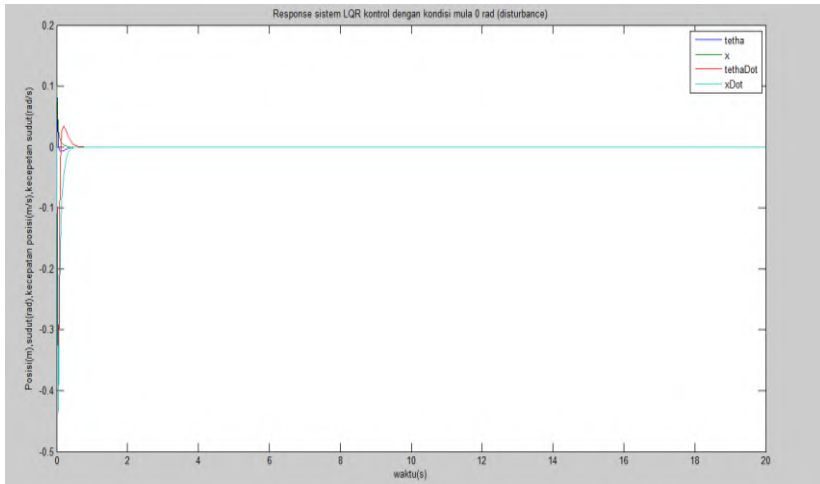
Gambar 4.7 Respon *tracking* sistem pada posisi robot menggunakan input dengan kondisi mula 0 rad



Gambar 4.8 Respon *tracking* sistem pada posisi robot menggunakan input dengan kondisi mula ± 0.2 rad

4.1.2 Hasil Simulasi Sistem dengan Gangguan Internal Robot

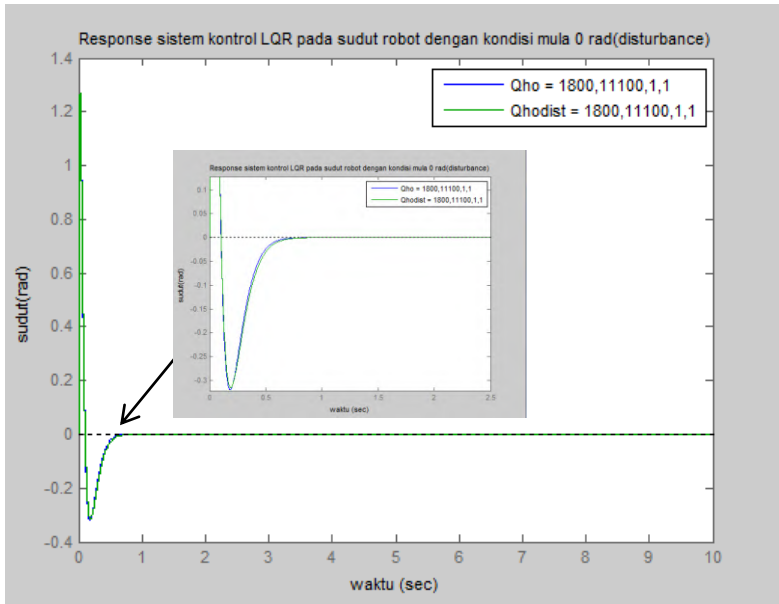
Kondisi mula ini tanpa menggunakan input dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg) dan panjang robot (0,062 m). Gangguan internal robot dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,44 kg) dan panjang robot (0,088 m). Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 1800, 11100, 1, 1) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dapat dilihat seperti Gambar 4.9:



Gambar 4.9 Respon sistem tanpa input dengan kondisi mula 0 rad (disturbance)

4.1.2.1 *Response sudut robot menggunakan input step*

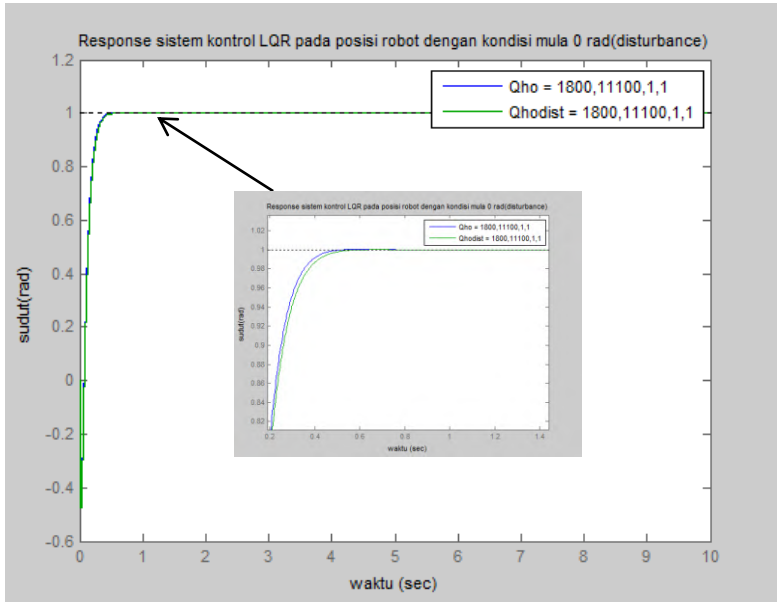
Kondisi mula ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg) dan panjang robot (0,062 m). Gangguan *internal* robot dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,44 kg) dan panjang robot (0,088 m). Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 1800, 11100, 1, 1) dan (R = 1). Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: (Q = 700, 22500, 1, 1) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dapat dilihat seperti Gambar 4.10:



Gambar 4.10 Respon sistem pada sudut robot menggunakan input dengan kondisi mula 0 rad

4.1.2.2 *Response posisi robot menggunakan input step*

Kondisi mula ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg), panjang robot (0,062 m) dan *state* output diinginkan ($y_d = [0 \ 1 \ 0 \ 0]$). Gangguan *internal* robot dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,44 kg), panjang robot (0,088 m) dan *state* output diinginkan ($y_d = [0 \ 1 \ 0 \ 0]$).. Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 1800, 11100, 1, 1) dan (R = 1). Sedangkan untuk kondisi mula ± 0.2 rad nilai pembobot Q dan R optimal adalah: (Q = 700, 22500, 1, 1) dan (R = 1). Hasil simulasi untuk kondisi mula 0 rad dapat dilihat seperti Gambar 4.11:



Gambar 4.11 Respon sistem pada posisi robot menggunakan input dengan kondisi mula 0 rad

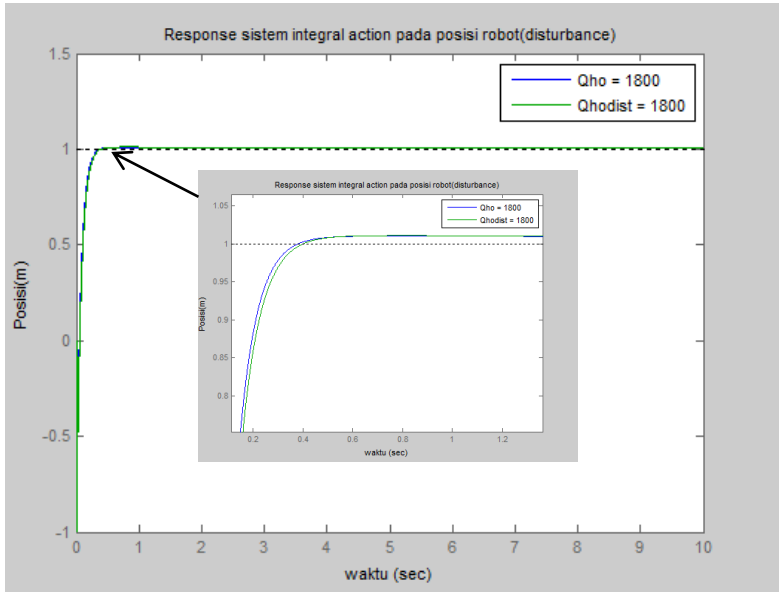
4.1.2.3 Integral Action Tracking Control

Pendekatan dasar di dalam umpan balik integral dibentuk dari *state* tidak lebih dari kontroler dengan perhitungan integral dari sinyal *error*, yang mana menggunakan teori dari umpan balik.

$$\frac{d}{dt} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix}$$

Kondisi mula ini menggunakan input *step* dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,51 kg), panjang robot (0,062 m) dan *state* output diinginkan ($y_d = [0 \ 1 \ 0 \ 0]$). Gangguan *internal* robot dengan *state* mula ($x_0 = [0.1 \ 0.1 \ 0 \ 0]$), beban robot (0,44 kg), panjang robot (0,088 m). Untuk kondisi mula 0 rad nilai pembobot Q dan R optimal yang digunakan adalah: (Q = 10, 0, 1800, 10) dan (R = 1).

Hasil simulasi untuk kondisi mula 0 rad dapat dilihat seperti Gambar 4.12 :



Gambar 4.12 Respon *tracking* sistem pada posisi robot menggunakan input dengan kondisi mula 0 rad

4.2 Hasil Implementasi

Dalam tahapan proses implementasi digunakan kondisi mula ± 0.2 radian. Sedangkan kondisi mula 0 radian, sinyal kontrol sangat kecil sehingga energi yang dikeluarkan untuk ke setimbangan sangat kecil. Untuk kondisi mula ± 0.2 radian maka didapatkan hasil *gain* K kontroler adalah :

$$K = [184.6081 \quad -105.3565 \quad 13.9866 \quad -18.6792] \text{ (kondisi mula } 0.2 \text{ rad)}$$

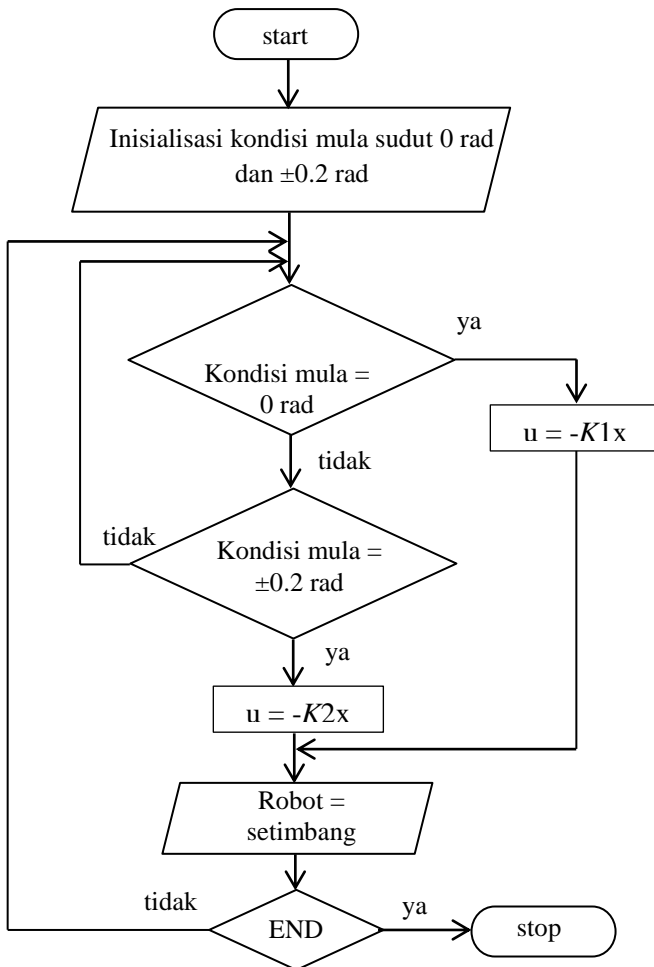
$$K_i = [772.1329 \quad 59.9641 \quad -44.7633 \quad -3.1623]$$

Maka sinyal kontrol ialah:

$$u = -Kx$$

didalam pengujian saya menggunakan nilai K kontroler sebagai acuan dalam menyusun program yang akan ditanam di dalam mikroprosesor / mikrokontroler .

- Pembuatan *flowchart* program mikrokontroler

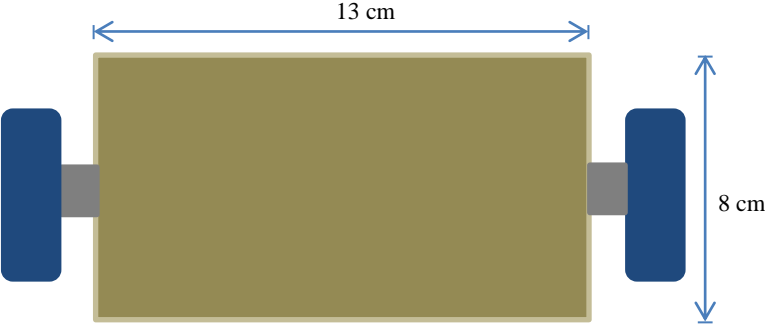


Hasil implementasi yang di dapat tidak sesuai dengan yang apa diharapkan, robot masih tidak dapat mempertahankan posisi tegak di daerah kerja 0 radian.

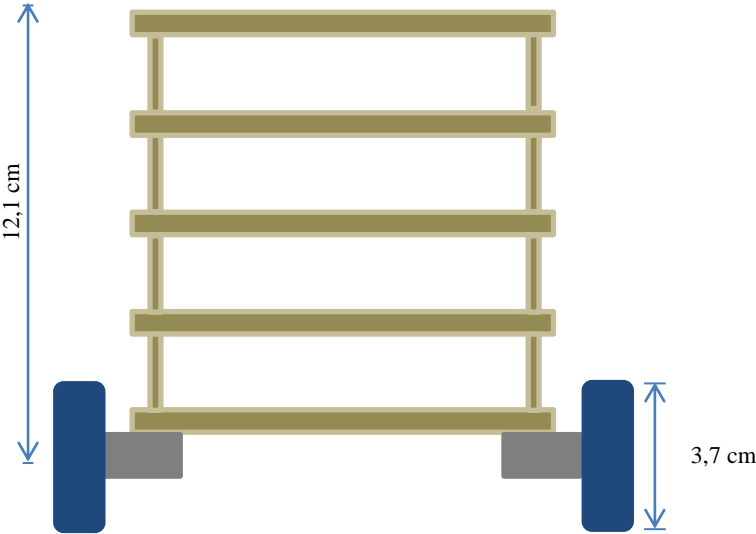
Halaman ini sengaja dikosongkan

LAMPIRAN 1

Konsep Robot



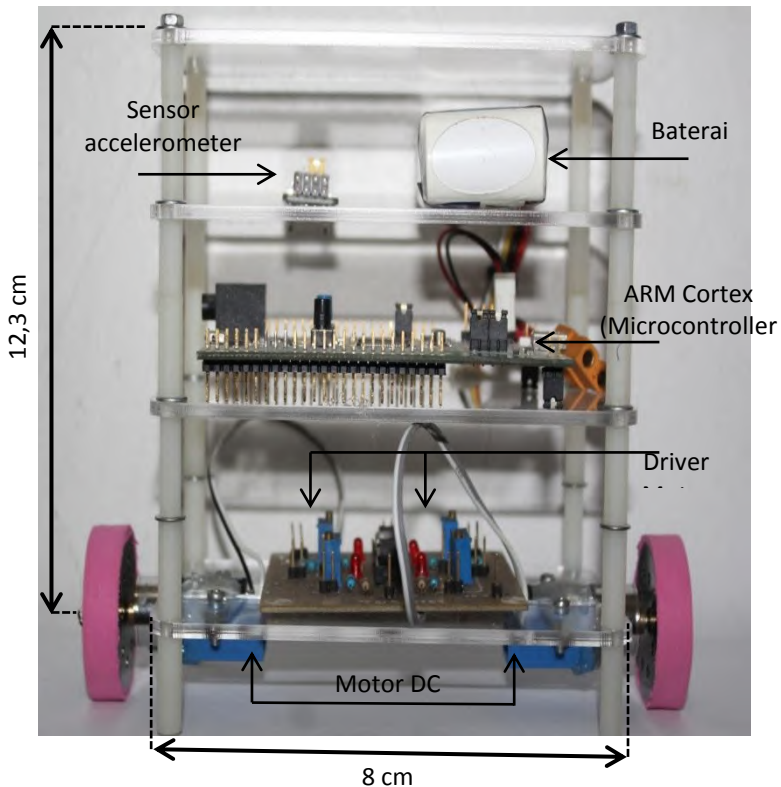
Gambar 1. Tampak atas



Gambar 2. Tampak Depan

Struktur Robot

Mengikuti teorema dari *inverted pendulum* untuk desain dan pengembangan dari *two wheeled inverted pendulum segway models*. Dalam desain chassis ini menggunakan 4 level tingkatan untuk menyusun komponen dan untuk representasi dari pendulum (Gambar 1). Komponen-komponen pendukung untuk sistem kerja robot model segway ini terdiri dari (baterai, mikrokontroler, sensor accelerometer & gyro, driver motor, motor DC).



Gambar 3. Chassis dari *inverted pendulum* dua roda model Segway

Spesifikasi Robot

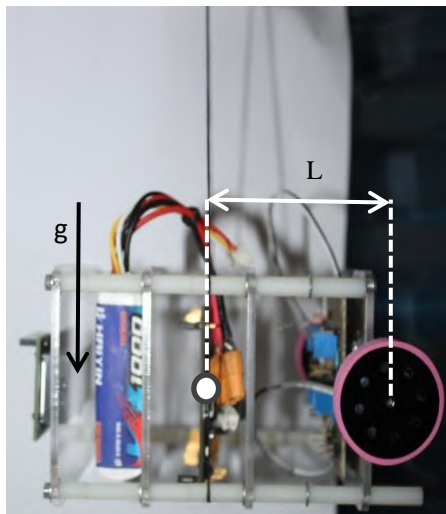
Persamaan linier dari robot inverted pendulum model *segway* berisi berbagai nilai para-meter fisik yang perlu dicari diantaranya berat *chassis*, berat roda, jari-jari roda, dan lain-lain. Parameter dicari dengan pengukuran terpisah. Hasil dari observasi parameter digunakan untuk model matematika dari plant, simulasi sistem dan rancangan program pada proses implementasi.

Berat *chassis* dan roda

Berat *chassis* dan berat roda diukur dengan timbangan dimana untuk berat *chassis* meliputi (pendulum dan motor DC), Sedangkan berat roda diukur jumlah total dari kedua roda (roda kanan dan roda kiri).

Pusat gravitasi

Untuk mencari nilai gravitasi dari robot pendulum terbalik model *segway* dengan cara menggantungkan beban terhadap sumbu horizontal. Jarak antara sumbu pusat roda dengan titik berat pusat gravitasi merupakan nilai yang dicari. Gambar 4, adalah ilustrasi untuk mencari nilai pusat gravitasi.



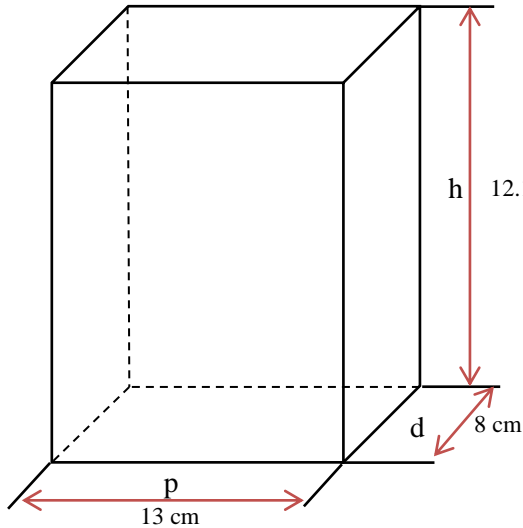
Gambar 4. Perhitungan pusat gravitasi robot model *Segway*

Momen inersia *chassis*

Chassis kami asumsikan sebagai kubus persegi panjang yang mana untuk mempermudah dalam proses perhitungan. Persamaan (1) merupakan perhitungan untuk mencari nilai rotasi momen inersia dari *chassis* robot.

$$j_c = \frac{1}{12} m_c (h^2 + d^2) \quad (1)$$

dimana: m_c ialah massa *chassis*
 h ialah tinggi *chassis*
 d ialah lebar *chassis*



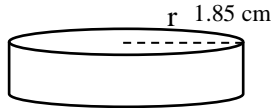
Gambar 5. Pengukuran model kubus

Momen inersia *wheel*

Kami asumsikan roda robot berbentuk silinder. Persamaan (2) merupakan perhitungan untuk mencari nilai rotasi momen inersia dari roda robot.

$$j_w = \frac{1}{2} m_w r^2 \quad (2)$$

dimana m_w dan R adalah massa dari roda dan jari-jari roda pada Gambar 6.



Gambar 6. Pengukuran model tabung

Torsi motor

$$\tau = F \cdot r$$

$$\tau_m = (m_w + m_c) \cdot g \cdot R$$

Pada Tabel 1 merupakan hasil dari proses menemukan nilai parameter robot pendulum terbalik beroda dua model *Segway*.

Tabel 1. Hasil parameter dari robot *inverted pendulum* model *segway*

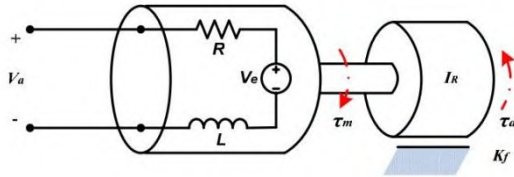
Nilai	Simbol	Parameter
$g = 9.81$	[m/s ²]	gravitasi bumi
$m_c = 0.51$	[kg]	berat <i>chassis</i>
$m_w = 0.02$	[kg]	berat roda
$j_c = 0.00089424$	[kgm ²]	momen inersia dari <i>chassis</i>
$j_w = 0.00003425$	[kgm ²]	momen inersia dari roda
$R = 0.123$	[m]	jarak antara pusat roda dengan tinggi ujung <i>chassis</i>
$r = 0.0185$	[m]	jari-jari roda
$p = 0.13$	[m]	panjang <i>chassis</i>
$h = 0.121$	[m]	tinggi <i>chassis</i>
$d = 0.08$	[m]	lebar <i>chassis</i>
$l = 0.062$	[m]	jarak pusat gravitasi dengan roda
$\tau_m = 0.6395$	[Nm]	torsi motor kanan dan kiri
$\tau_m^{kiri} = 0.31975$	[Nm]	torsi motor kiri
$\tau_m^{kanan} = 0.31975$	[Nm]	torsi motor kanan

Halaman ini sengaja dikosongkan

LAMPIRAN 2

- ❖ Penurunan model dinamis dari sistem dibagi 3 bagian:
 1. Model dinamis motor DC
 2. Model dinamis roda
 3. Model dinamis chassis / pendulum

Model dinamis motor DC :



Gambar 1. Diagram dari motor DC

Hukum kirchoff tegangan:

$$V_{in} - E_{ggl} = R \cdot i + L \frac{di}{dt} \quad (1)$$

Motor searah menghasilkan torsi untuk menggerakkan chassis robot agar dapat berdiri tegak & seimbang. Torsi yang dihasilkan akan memutar armature dan menetralkan gaya yang dibangkitkan oleh motor.

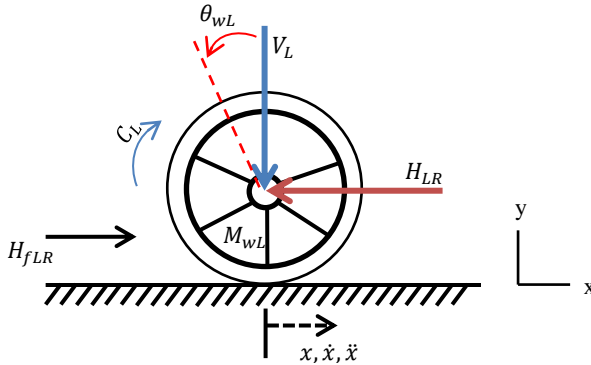
$$\tau_m = K_{TM} \cdot \phi i \quad (2)$$

- K_{TM} = konstanta torsi motor (Nm/A)
- Φ = fluks per kutub pada kumparan medan (wb)
- i = arus armature (A)
- τ_m = torsi motor (Nm)

Fluks pada motor searah dapat diabaikan karna menggunakan magnet permanent

$$\tau_m = K_{TM} \cdot i \quad (3)$$

Model dinamis roda motor :



Gambar 2. Diagram kebebasan dari roda

Persamaan *State* robot didapat dari analisa gaya sesuai dengan hukum Newton. Persamaan gaya translasi pada bidang horizontal dapat ditulis sebagai berikut.

$$\sum F_x = m \cdot a_x$$

$$M_{wLR} \ddot{x} = H_{fLR} - H_{LR} \quad (4)$$

Dimna :

M_{wLR} = massa roda kiri dan kanan (kg)

H_{fLR} = gaya friksi horizontal pada roda kiri dan kanan (N)

Hukum gerak newton, penjumlahan momen gaya di pusat roda.

$$\sum M_0 = J_{wRL} \ddot{\theta}_{wLR} \quad (5)$$

$$C_{LR} - H_{fLR} r = J_{wRL} \ddot{\theta}_{wLR} \quad (6)$$

karna putaran roda digerakkan oleh torsi motor $C_L = C_R = \tau_m$ jadi persamaan menjadi sebagai berikut

$$\tau_m - H_{fLR} r = J_{wRL} \ddot{\theta}_{wLR}$$

$$H_{fLR} = \frac{\tau_m}{r} - \frac{J_{wRL} \ddot{\theta}_{wLR}}{r} \quad (8)$$

Dimana :

r = jarak beban terhadap pusat rotasi (m)

τ_m = torsi motor

J_{wRL} = momen inersia roda kanan dan kiri

Selanjutnya persamaan (4) di distribusikan ke persamaan (8) maka diperoleh.

$$M_{wLR} \ddot{x} = \frac{\tau_m}{r} - \frac{J_{wRL} \ddot{\theta}_{wLR}}{r} - H_{LR} \quad (9)$$

Karna gerak linier muncul pada roda, maka gerak rotasi angular dapat ditransformasikan menjadi gerak linier dengan transformasi berikut.

$$x = \theta \cdot r, \quad \dot{x} = \dot{\theta} \cdot r, \quad \ddot{x} = \ddot{\theta} \cdot r$$

atau

$$\theta = \frac{x}{r}, \quad \dot{\theta} = \frac{\dot{x}}{r}, \quad \ddot{\theta} = \frac{\ddot{x}}{r} \quad (10)$$

transformasikan persamaan (10) ke persamaan (9) maka persamaan akan menjadi seperti persamaan :

a. Untuk roda kanan

$$M_{wR} \ddot{x} = \frac{\tau_m}{r} - \frac{J_{wR} \ddot{x}_{wR}}{r} - H_R \quad (11)$$

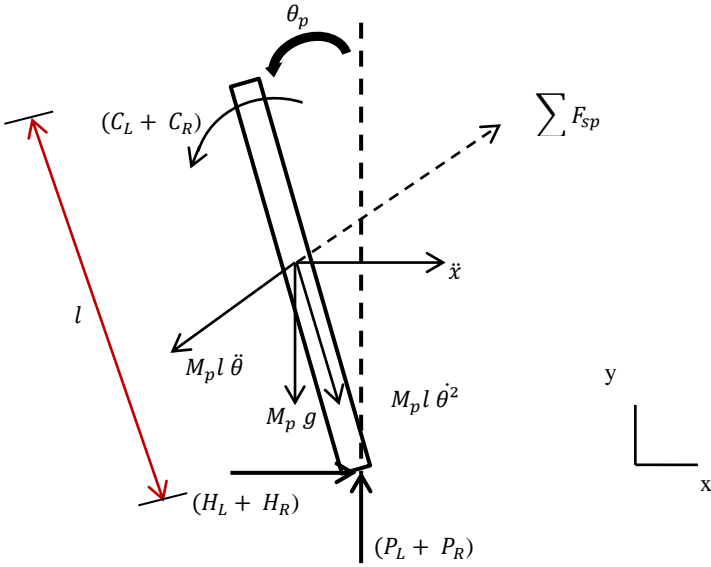
b. Untuk roda kiri

$$M_{wL} \ddot{x} = \frac{\tau_m}{r} - \frac{J_{wL} \ddot{\omega}_{wLR}}{r} - H_L \quad (12)$$

Jumlahan dari persamaan (11) dan (12),

$$2\left(M_w + \frac{J_w}{r^2}\right) \ddot{x} = \frac{2\tau_m}{r} - (H_L + H_R) \quad (13)$$

Model dinamis chassis robot :



Gambar 3. Diagram kebebasan *body chassis*

Menggunakan hukum Newton. Penjumlahan gaya pendulum terhadap bidang horizontal dapat ditulis sebagai berikut.

$$\begin{aligned} \sum F_x &= M_p \ddot{x} \\ (H_L + H_R) - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p &= M_p \ddot{x} \end{aligned} \quad (14)$$

Dapat di tulis juga :

$$(H_L + H_R) = M_p \ddot{x} + M_p l \ddot{\theta} \cos \theta_p - M_p l \dot{\theta}^2 \sin \theta_p \quad (15)$$

Untuk penjumlahan gaya yang tegak lurus pada pendulum (F_{sp}) persamaan dapat diturunkan,

$$\begin{aligned} \sum F_{sp} &= M_p \ddot{x} \cos \theta_p \\ (H_L + H_R) \cos \theta_p + (P_L + P_R) \sin \theta_p - M_p g \sin \theta_p - M_p l \ddot{\theta} &= \\ M_p \ddot{x} \cos \theta_p & \quad (16) \end{aligned}$$

Untuk menjumlahkan momen disekitar pusat massa dari pendulum (M_o) dapat diturunkan persamaan,

$$\begin{aligned} \sum M_o &= J_p \ddot{\theta}_p \\ -(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p - (C_L + C_R) &= J_p \ddot{\theta}_p \quad (17) \end{aligned}$$

Keterangan:

$$\begin{aligned} J_p &= \text{inersia dari sistem (scooter + rider)} \\ P_L + P_R &= \text{gaya reaksi diantara roda dan chassis} \\ C_L \text{ \& } C_R &= \text{torsi dari motor } (\tau_m) \end{aligned}$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p - 2 \tau_m = J_p \ddot{\theta}_p \quad (18)$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p = J_p \ddot{\theta}_p + 2 \tau_m$$

Dengan mengalikan (16) dengan $-l$ maka,

$$[-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p] + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta} = -M_p l \ddot{x} \cos \theta_p \quad (20)$$

Kemudian substitusikan persamaan (20) ke persamaan (19),

$$J_p \ddot{\theta}_p + 2 \tau_m + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta} = -M_p l \ddot{x} \cos \theta_p \quad (21)$$

Eliminasi gaya ($H_L + H_R$) dari dinamis motor, persamaan (15) di substitusikan ke persamaan (13),

$$2\left(M_w + \frac{J_w}{r^2}\right)\ddot{x} = \frac{2\tau_m}{r} - M_p\ddot{x} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p \quad (22)$$

Model dinamis robot :

Persamaan (21),

$$\begin{aligned} J_p \ddot{\theta}_p + 2 \tau_m + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta} &= -M_p l \ddot{x} \cos \theta_p \\ (J_p + M_p l^2) \ddot{\theta}_p + 2 \tau_m + M_p g l \sin \theta_p &= -M_p l \ddot{x} \cos \theta_p \\ \ddot{\theta}_p &= \frac{-M_p l \ddot{x} \cos \theta_p - 2 \tau_m - M_p g l \sin \theta_p}{(J_p + M_p l^2)} \end{aligned} \quad (23)$$

Persamaan (22),

$$\begin{aligned} 2\left(M_w + \frac{J_w}{r^2}\right)\ddot{x} &= \frac{2\tau_m}{r} - M_p\ddot{x} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p \\ (2M_w + \frac{2J_w}{r^2} + M_p) \ddot{x} &= \frac{2\tau_m}{r} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p \\ \ddot{x} &= \frac{\frac{2\tau_m}{r} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p}{2M_w + \frac{2J_w}{r^2} + M_p} \end{aligned} \quad (24)$$

Sebelum proses lebih lanjut, maka harus melinierisasikan persamaan state. Ada dua nilai kesetimbangan didalam pendulum ialah saat $\theta = 0$ (stabil) dan $\theta = \pi$ (tidak stabil). Dari persamaan (23) dan (24) dapat di-linierisasikan dengan mengasumsikan $\theta_p = \pi$, dimana merupakan sudut kecil terhadap arah vertikal. Penyederhanaan ini digunakan untuk mendapatkan permodelan linier pada persamaan non-linier untuk memperoleh kontrol *state-space* sehingga dapat di implementasikan. Untuk itu,

$$\begin{aligned} \cos \theta_p &= \cos (\pi) + (\pi - \theta)(-\sin \pi) = -1 \\ \sin \theta_p &= \sin (\pi) + (\pi - \theta)(\cos \pi) = \theta - \pi \end{aligned} \longrightarrow \text{Merupakan koordinat baru dari } \pi - \theta$$

maka, karna terdapat koordinat baru jadi

$$\bar{\theta} = \theta - \pi$$

Pendulum dalam kondisi sejajar dengan sumbu y, dengan pergerakan θ_p *counter clockwise*.

$$\sin \theta_p = \pi - (\bar{\theta} + \pi) = -\bar{\theta}$$

$\bar{\theta} = \theta$ dengan catatan bahwa $\bar{\theta}$ adalah nilai θ baru.

$$\sin \theta_p = -\theta$$

$$\text{Jadi } \cos \theta_p = -1, \sin \theta_p = -\theta, \left(\frac{d\theta_p}{dt} \right)^2 = 0$$

dengan menggunakan aturan diatas, maka :

$$(J_p + M_p l^2) \ddot{\theta}_p + 2 \tau_m + M_p g l \sin \theta_p = -M_p l \ddot{x} \cos \theta_p$$

$$(J_p + M_p l^2) \ddot{\theta} + 2 \tau_m + M_p g l \theta = M_p l \ddot{x} \quad (25)$$

$$\ddot{\theta} = \frac{M_p l \ddot{x} - 2 \tau_m - M_p g l \theta}{(J_p + M_p l^2)} \quad (26)$$

$$\left(2M_w + \frac{2J_w}{r^2} + M_p \right) \ddot{x} = \frac{2 \tau_m}{r} - M_p l \ddot{\theta} \cos \theta_p + M_p l \dot{\theta}^2 \sin \theta_p$$

$$\left(2M_w + \frac{2J_w}{r^2} + M_p \right) \ddot{x} = \frac{2 \tau_m}{r} + M_p l \ddot{\theta} \quad (27)$$

$$\ddot{x} = \frac{\frac{2 \tau_m}{r} + M_p l \ddot{\theta}}{2M_w + \frac{2J_w}{r^2} + M_p} \quad (28)$$

$$\text{Jika } \alpha = 2M_w + \frac{2J_w}{r^2} + M_p$$

maka,

$$\ddot{x} = \frac{\frac{2 \tau_m}{r} + M_p l \ddot{\theta}}{\alpha} = \frac{2 \tau_m}{r \alpha} + \frac{M_p l \ddot{\theta}}{\alpha} \quad (29)$$

Halaman ini sengaja dikosongkan

LAMPIRAN 3

Dalam persamaan state terlebih dahulu mendefinisikan variable,

$$x_1 = \theta, \quad x_2 = \dot{\theta}, \quad x_3 = \ddot{\theta}, \quad x_4 = \dddot{\theta}$$

Diperoleh persamaan berdasarkan persamaan (25) dan persamaan (27)

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = \frac{M_p g l \theta}{\beta - \left(\frac{M_p^2 l^2}{\alpha}\right)} + \frac{\frac{2 M_p l}{r \alpha} - 2}{\beta - \left(\frac{M_p^2 l^2}{\alpha}\right)} \tau_m$$

$$\dot{x}_4 = \frac{(M_p^2 g l^2) \theta}{\beta \alpha - M_p^2 l^2} + \frac{2 \beta - 2 M_p l r}{r (\beta \alpha - M_p^2 l^2)} \tau_m$$

Persamaan state variable

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{M_p g l \theta}{\beta - \left(\frac{M_p^2 l^2}{\alpha}\right)} & 0 & 0 & 0 \\ \frac{(M_p^2 g l^2) \theta}{\beta \alpha - M_p^2 l^2} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\frac{2 M_p l}{r \alpha} - 2}{\beta - \left(\frac{M_p^2 l^2}{\alpha}\right)} \\ \frac{2 \beta - 2 M_p l r}{r (\beta \alpha - M_p^2 l^2)} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -203.8267 & 0 & 0 & 0 \\ 8.5917 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 370.7894 \\ 214.9381 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Halaman ini sengaja dikosongkan

LAMPIRAN 4

`%----- model untuk 0 radian -----%`

`% Tugas Akhir Tracking Kontrol Robot Invers
Pendulum Beroda Dua Model Segway
% Farid Zainul Hasan - 2214 105 026 (Teknik
Sistem Pengaturan)`

`% Model Dinamis Robot
%
% Model dinamis robot, ditulis dalam state space
%
% matrik sistem
A = [0 0 1 0; 0 0 0 1; 203.82 0 0 0; 8.59 0 0
0];
B = [0;0;370.78;213.93];
C = [1 0 0 0; 0 1 0 0];
D = [0];`

`%
% konstruksi input dan output dari response step
xy posisi
% xd dan yd memiliki hubungan state bahwa state
equilibrium yang diinginkan
% dari sistem. matrik Cy berhubungan dengan
Output.`

`% $\dot{x} = Ax + Bu \Rightarrow \dot{x} = (A-BK)x + K x_d$
% $u = -K(x-x_d) \quad y = Cx$`

`% dinamis dari close loop dapat disimulasikan
dimana saya menggunakan
% "step", dengan $K \cdot x_d$ sebagai input vektor
(asumsi bahwa "input" adalah
% terukur, jadi x_d berhubungan dengan steady
state yang diinginkan.`

```

xd = [1;0;0;0]; Cx = [1 0 0 0];
yd = [0;1;0;0]; Cy = [0 1 0 0];

%
% LQR desain
%

% dimulai dengan a pembobot diagonal
Q1 = diag([1800,11100,1,1]);
R1a = 1;
K1a = lqr(A,B,Q1,R1a);

Abaru=A-B*K1a;
damp(Abaru)

x0= [0.1 0.1 0 0]';

t=(0:0.05:20);
u=zeros(size(t));
[y,x]=lsim(Abaru,B,C,D,u,t,x0);
figure(1); plot(t,x)
title('Response sistem LQR kontrol dengan
kondisi mula 0 rad');
xlabel('waktu(sec)');
ylabel('Posisi(m), sudut(rad), kecepatan
posisi(m/s), kecepatan sudut(rad/s)');
legend('tetha','x','tethaDot','xDot');

% close loop : xdot = Ax + B K (x -xd)

H1ax = ss(A-B*K1a,B(:,1)*K1a(1,:)*xd,Cx,0);
H1ay = ss(A-B*K1a,B(:,1)*K1a(1,:)*yd,Cy,0);

figure(2); step (H1ax,10);%H1bx,H1cx,H1ex
title('Response sistem kontrol LQR pada sudut
robot dengan kondisi mula 0 rad');
xlabel('waktu');

```

```

ylabel('sudut(rad)');
legend('','Qho =
1800,11100,1,1');%,'900,500,1,1','2400,1200,1,1'
,'3600,1700,1,1','6300,2600,1,1'

figure(3); step (H1ay,10);%H1by,H1cy,H1ey,H1fy,
title('Response sistem kontrol LQR pada posisi
robot dengan kondisi mula 0 rad');
xlabel('waktu');
ylabel('posisi(m)');
legend('Qho = 1800,11100,1,1');

% Pembobotan Output

Q2 = [Cx;Cy]'*[Cx;Cy];
R2 = 0.1 *diag([1]);
K2 = lqr(A,B,Q2,R2);

H2x = ss(A-B*K2,B(:,1)*K2(1,:)*xd,Cx,0);
H2y = ss(A-B*K2,B(:,1)*K2(1,:)*yd,Cy,0);
%figure(3); step (H2x,H2y,10);
% xlabel('waktu(s)');
% ylabel('Posisi(m) dan sudut(rad)');
% legend('x = sudut','y = posisi');

% pembobotan secara fisik

% untuk x error 1 cm, dan y error 10 cm.
mencoba menjaga sudut tidak
% kurang dari 5 derajat dalam membuat
penyesuaian.

Q3 = diag([1800, 11100, 1, 1]);
R3 = diag([1]);
K3 = lqr(A,B,Q3,R3);

H3x = ss(A-B*K3,B(:,1)*K3(1,:)*xd,Cx,0);
H3y = ss(A-B*K3,B(:,1)*K3(1,:)*yd,Cy,0);

```



```

%figure(4); step (H3x,H3y,10);
% xlabel('waktu(s)');
% ylabel('Posisi(m) dan sudut(rad)');
% legend('x = sudut','y = posisi');

% kontrol kecepatan

% memodifikasi sistem sehingga dikontrol
kecepatan dari sistem terhadap arah
sumbu x. untuk arah sumbu y diabaikan.

% digunakan hukum kontrol  $u = -K(x-x_d) + \dot{u}_d$ 

% dimana ekstrak matrix "theta, thdot, xdot"

Av = A([1 3 4],[1 3 4]);
Bv = B([1 3 4],1);
Cv = [0 0 1];
Dv = 0;

% Desain #1: Integral aksi

% menambahkan state baru pada sistem dengan
diberikan dari  $\dot{x}_{id} = v - v_d$ .
% dikonstruksikan hukum kontrol dari komputasi
sebuah gain LQR untuk
% sistem augmented.

Ai = [Av, [0;0;0]; [Cv,0]];
Bi = [Bv;0];
Ci = [Cv,0];
Di = Dv;

% desain teorema feedback, memasukkan pembobot
pada integral Action error

Qi = diag([10, 0, 1800, 10]);
Ri = 1;

```

```

Ki = lqr (Ai, Bi, Qi, Ri);

%state yang diinginkan (augmented)
xid =[0; 0; 1; 0];

% close loop sistem (ditambahkan integrator)
Hi = ss (Ai-Bi*Ki,Bi*Ki*xid - [0; 0; 0;
Ci*xid],Ci,0);
%
%

figure(4); step(Hi,10); %Hia,Hib,Hic,Hid,Hie,
title('Response sistem integral action pada
posisi robot dengan kondisi mula 0 rad');
xlabel('waktu');
ylabel('Posisi(m)');
legend('Qho = 10,0,1800,10');

%----- model untuk  $\pm 0.2$  radian -----%

% Tugas Akhir Tracking Kontrol Robot Invers
Pendulum Beroda Dua Model Segway
% Farid Zainul Hasan - 2214 105 026 (Teknik
Sistem Pengaturan)

% Model Dinamis Robot
%
% Model dinamis robot, ditulis dalam state space
%
% matrik sistem
A = [0 0 1 0; 0 0 0 1; 181.113 0 0 0; 10.3106 0
0 0];
B = [0;0;403.88;271.718];
C = [1 0 0 0; 0 1 0 0];

```

```

D = [0];

%
% konstruksi input dan output dari response step
xy posisi
% xd dan yd memiliki hubungan state bahwa state
equilibrium yang diinginkan
% dari sistem. matrik Cy berhubungan dengan
Output.

%  $\dot{x} = Ax + Bu \Rightarrow \dot{x} = (A-BK)x + K x_d$ 
%  $u = -K(x-x_d) \quad y = Cx$ 

% dinamis dari close loop dapat disimulasikan
dimana saya menggunakan
% "step", dengan  $Kx_d$  sebagai input vektor
(asumsi bahwa "input" adalah
% terukur, jadi  $x_d$  berhubungan dengan steady
state yang diinginkan.

xd = [1;0;0;0]; Cx = [1 0 0 0];
yd = [0;1;0;0]; Cy = [0 1 0 0];

%
% LQR desain
%

% dimulai dengan a pembobot diagonal
Q1 = diag([1800,11100,1,1]); %2*pi/24
R1a = 1;
%R1a = 0.1 * diag([1100]);
K1a = lqr(A,B,Q1,R1a);

Abaru=A-B*K1a;
damp(Abaru)

x0= [0.1 0.1 0 0]';

```

```

t=(0:0.05:20);
u=zeros(size(t));
[y,x]=lsim(Abaru,B,C,D,u,t,x0);
figure(1); plot(t,x)
title('Response sistem LQR kontrol dengan
kondisi mula 0 rad');
xlabel('waktu(sec)');
ylabel('Posisi(m),sudut(rad),kecepatan
posisi(m/s),kecepatan sudut(rad/s)');
legend('tetha','x','tethaDot','xDot');
% close loop : xdot = Ax + B K (x -xd)

H1ax = ss(A-B*K1a,B(:,1)*K1a(1,:)*xd,Cx,0);
H1ay = ss(A-B*K1a,B(:,1)*K1a(1,:)*yd,Cy,0);
figure(2); step (H1ax,10);%H1bx,H1cx,H1ex,
title('Response sistem kontrol LQR pada sudut
robot dengan kondisi mula 0 rad');
xlabel('waktu');
ylabel('sudut(rad)');
legend('Qho =
1800,11100,1,1');%,'900,500,1,1','2400,1200,1,1'
,'3600,1700,1,1','6300,2600,1,1'

figure(3); step (H1ay,10);%H1by,H1cy,H1ey,H1fy,
title('Response sistem kontrol LQR pada posisi
robot dengan kondisi mula 0 rad');
xlabel('waktu');
ylabel('posisi(m)');
legend('Qho = 1800,11100,1,1');%,'Qho =
900,500,1,1','Qho = 2400,1200,1,1','Qho =
3600,1700,1,1','Qho = 6300,2600,1,1'

% Pembobotan Output

Q2 = [Cx;Cy]'*[Cx;Cy];
R2 = 0.1 *diag([1]);
K2 = lqr(A,B,Q2,R2);

H2x = ss(A-B*K2,B(:,1)*K2(1,:)*xd,Cx,0);

```

```
H2y = ss(A-B*K2,B(:,1)*K2(1,:)*yd,Cy,0);
%figure(3); step (H2x,H2y,10);
%legend('x = sudut','y = posisi');
```

```
% pembobotan secara fisik
```

```
% untuk x error 1 cm, dan y error 10 cm.
mencoba menjaga sudut tidak
% kurang dari 5 derajat dalam membuat
penyesuaian.
```

```
Q3 = diag([1800, 11100, 1, 1]);
R3 = diag([1]);
K3 = lqr(A,B,Q3,R3);
```

```
H3x = ss(A-B*K3,B(:,1)*K3(1,:)*xd,Cx,0);
H3y = ss(A-B*K3,B(:,1)*K3(1,:)*yd,Cy,0);
%figure(4); step (H3x,H3y,10);
%legend('x = sudut','y = posisi');
```

```
% kontrol kecepatan
% memodifikasi sistem sehingga dikontrol
kecepatan dari sistem terhadap arah
% sumbu x. untuk arah sumbu y diabaikan.
% digunakan hukum kontrol  $u = -K(x-xd) + ud$ 
% dimana ekstrak matrix "theta, thdot, xdot"
```

```
Av = A([1 3 4],[1 3 4]);
Bv = B([1 3 4],1);
Cv = [0 0 1];
Dv = 0;
```

```
% Desain #1: Integral aksi
```

```
% menambahkan state baru pada sistem dengan
diberikan dari  $xidot = v - vd$ .
% dikonstruksikan hukum kontrol dari komputasi
sebuah gain LQR untuk
```

```

% sistem augmented.

Ai = [Av, [0;0;0];[Cv,0]];
Bi = [Bv;0];
Ci = [Cv,0];
Di = Dv;

% desain teorema feedback, memasukkan pembobot
pada integral Action error

Qi = diag([10, 0, 1960, 10]);
Ri = 1;
Ki = lqr (Ai, Bi, Qi, Ri);

%state yang diinginkan (augmented)
xid =[0; 0; 1; 0];

% close loop sistem (ditambahkan integrator)
Hi = ss (Ai-Bi*Ki,Bi*Ki*xid - [0; 0; 0;
Ci*xid],Ci,0);
%

figure(4); step(Hi,10); %Hia,Hib,Hic,Hid,Hie,
title('Response sistem integral action pada
posisi robot dengan kondisi mula 0.2 rad');
xlabel('waktu');
ylabel('Posisi(m)');
legend('Qho = 10, 0, 1960, 10');

%- model untuk 0 radian ada gangguan internal-%

%untuk gangguan internal (l = 0.088 m, Mp =
0.440 Kg dan L = 0.142 m)

Mp = 0.51;
g = 9.81;

```

```

l = 0.062;
Jp = 0.0008942;
Mw = 0.02;
r = 0.0185;
Jw = 0.00003425;

Beta = Jp + Mp*l^2;
Alpha = (2*Mw) + (2*Jw/r^2) + Mp;

A = [0
      0      1      0;
      0      0      1;
      (0.25*Mp*g*l*Alpha) / ((Beta*Alpha) -
      ((0.965^2*Mp^2*l^2)))      0      0
      0;
      (0.241*Mp^2*g*l^2) / (((Beta*Alpha) -
      (0.965^2*Mp^2*l^2)))      0      0
      0];

B = [0;
      0;
      ((1.93*Mp*l*Alpha) -
      2*r*Alpha^2) / ((r*Alpha) * ((Beta*Alpha) -
      (0.965^2*Mp^2*l^2)));
      ((2*Beta^2) -
      (1.93*Mp*l*r*Beta)) / ((r*Beta) * ((Beta*Alpha) -
      (0.965^2*Mp^2*l^2)))]];

C = [1 0 0 0; 0 1 0 0];

D = [0];

% konstruksi input dan output dari response step
xy posisi
% xd dan yd memiliki hubungan state bahwa state
equilibrium yang diinginkan

```

```

% dari sistem. matrik Cy berhubungan dengan
Output.

%  $\dot{x} = Ax + Bu \Rightarrow \dot{x} = (A-BK)x + K x_d$ 
%  $u = -K(x-x_d) \quad y = Cx$ 

% dinamis dari close loop dapat disimulasikan
dimana saya menggunakan
% "step", dengan  $K \cdot x_d$  sebagai input vektor
(asumsi bahwa "input" adalah
% terukur, jadi  $x_d$  berhubungan dengan steady
state yang diinginkan.

xd = [1;0;0;0]; Cx = [1 0 0 0];
yd = [0;1;0;0]; Cy = [0 1 0 0];

%
% LQR desain
%

% dimulai dengan a pembobot diagonal
Q1 = diag([700,22500,1,1]);
R1a = 1;

K1a = lqr(A,B,Q1,R1a);

Abaru=A-B*K1a;
damp(Abaru)

x0= [0.1 0.1 0 0]';

t=(0:0.05:20);
u=zeros(size(t));
[y,x]=lsim(Abaru,B,C,D,u,t,x0);
figure(6); plot(t,x)
title('Response sistem LQR kontrol dengan
kondisi mula 0.2 rad');

```



```

xlabel('waktu(sec)');
ylabel('Posisi(m), sudut(rad), kecepatan  
posisi(m/s), kecepatan sudut(rad/s)');
legend('tetha', 'x', 'tethaDot', 'xDot');
% close loop : xdot = Ax + B K (x -xd)

Hlax = ss(A-B*K1a,B(:,1)*K1a(1,:)*xd,Cx,0);
Hlay = ss(A-B*K1a,B(:,1)*K1a(1,:)*yd,Cy,0);
figure(1); step (Hlax,Hlay,10);
xlabel('waktu');
ylabel('Posisi(m/s) dan sudut(rad/s)');
legend('x = sudut', 'y = posisi');

% Melihat perbedaan pembobot input
Q1b = diag([700,22500,1,1]);
K1b = lqr(A,B,Q1b,R1a);
H1bx = ss(A-B*K1b,B(:,1)*K1b(1,:)*xd,Cx,0);
H1by = ss(A-B*K1b,B(:,1)*K1b(1,:)*yd,Cy,0);

figure(2); step (H1bx,10);
title('Response sistem kontrol LQR pada sudut  
robot dengan kondisi mula 0.2 rad');
xlabel('waktu');
ylabel('sudut(rad)');
legend('Qho = 700,22500,1,1');

figure(7); step (H1by,10);
title('Response sistem kontrol LQR pada posisi  
robot dengan kondisi mula 0.2 rad');
xlabel('waktu');
ylabel('posisi(m)');
legend('Qho = 700,22500,1,1');

% Pembobotan Output
Q2 = [Cx;Cy]'*[Cx;Cy];
R2 = diag([1]);

```

```

K2 = lqr(A,B,Q2,R2);

H2x = ss(A-B*K2,B(:,1)*K2(1,:)*xd,Cx,0);
H2y = ss(A-B*K2,B(:,1)*K2(1,:)*yd,Cy,0);
figure(3); step (H2x,H2y,10);
xlabel('waktu');
ylabel('Posisi(m) dan sudut(rad)');
legend('x = sudut','y = posisi');

% pembobotan secara fisik
% untuk x error 1 cm, dan y error 10 cm.
mencoba menjaga sudut tidak
% kurang dari 5 derajat dalam membuat
penyesuaian.

Q3 = diag([700,22500,1,1]);
R3 = diag([1]);
K3 = lqr(A,B,Q3,R3);

H3x = ss(A-B*K3,B(:,1)*K3(1,:)*xd,Cx,0);
H3y = ss(A-B*K3,B(:,1)*K3(1,:)*yd,Cy,0);
figure(4); step (H3x,H3y,10);
xlabel('waktu');
ylabel('Posisi(m) dan sudut(rad)');
legend('x = sudut','y = posisi');

% kontrol kecepatan
% memodifikasi sistem sehingga dikontrol
kecepatan dari sistem terhadap arah
% sumbu x. untuk arah sumbu y diabaikan.
% digunakan hukum kontrol  $u = -K(x-xd) + \dot{u}_d$ 
% dimana ekstrak matrix "theta, thdot, xdot"

Av = A([1 3 4],[1 3 4]);
Bv = B([1 3 4],1);
Cv = [0 0 1];
Dv = 0;

```

```

% Desain #1: Integral aksi

% menambahkan state baru pada sistem dengan
diberikan dari xidot = v - vd.
% dikonstruksikan hukum kontrol dari komputasi
sebuah gain LQR untuk
% sistem augmented.

Ai = [Av, [0;0;0];[Cv,0]];
Bi = [Bv;0];
Ci = [Cv,0];
Di = Dv;

% desain teorema feedback, memasukkan pembobot
pada integral Action error

Qi = diag([10,0,2200,10]);
Ri = 1;
Ki = lqr (Ai, Bi, Qi, Ri);

%state yang diinginkan (augmented)
xid=[0; 0; 1; 0];

% close loop sistem (ditambahkan integrator)
Hi = ss(Ai-Bi*Ki,Bi*Ki*xid - [0; 0; 0;
Ci*xid],Ci,0);
%

figure(5); step(Hi,10);
title(' Response sistem integral action pada
posisi robot kondisi mula 0.2 rad');
xlabel('waktu');
ylabel('Posisi(m)');
legend('Qho = 10,0,2200,10');

```

BAB 5

KESIMPULAN

5.1 Kesimpulan

Dalam proses perancangan, simulasi dan implementasi mengenai *tracking* kontrol robot pendulum terbalik beroda dua model segway, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Performansi sistem kontrol dengan pendekatan metode *Linier Quadratic Regulator* dipengaruhi oleh nilai pembobot matrik **Q** dan **R**. Nilai pembobot matrik **Q** ([1800,11100,1,1]) dan **R** ([1]) yang diinginkan akan memberikan respon dan waktu yang lebih baik.
2. Hasil simulasi *Linear Quadratic Regulator* dengan pendekatan *integral action* menunjukkan bahwa posisi robot dapat mengikuti atau *men-track* sinyal referensi *step* yang diberikan.
3. Simulasi serta implementasi dalam tugas akhir ini menunjukkan perbedaan dalam hasil yang dicapai. Untuk simulasi bahwa posisi robot mampu mengikuti sinyal referensi *step*, sedangkan sudut robot dapat dipertahankan di sekitar 0 rad dengan *settling time* 2 detik. Sedangkan proses implementasi posisi robot tidak dapat mengikuti sinyal *reference* dan sudut robot tidak dapat mempertahankan di 0 radian.

5.2 Saran

Dalam kontrol *tracking* LQR (*Linear Quadratic Regulator*) dengan metode *integral action* sebaiknya untuk simulasi diuji dengan menggunakan sinyal referensi *sinus* dan *square-wave*. Untuk perkembangan berikutnya, dapat diujikan menggunakan sinyal referensi *sinus*, *square-wave*, dan *step* dengan kontroler lain sehingga robot tidak hanya dapat mengikuti sinyal referensi tapi dapat stabil di 3sinyal referensi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Chi Ooi, Rich., “*Balancing a Two-Wheeled Autonomous Robot*”, Final Year Thesis, The University of Western Australia School of Mechanical Engineering, 2003.
- [2] Zhou, Weiqian (Viking)., “*Platform for ergonomic steering methods investigation of Segway-style Balancing scooters*”, Final Year Thesis, The University of WAIKATO in physics and Electronic engineering, 2008.
- [3] Memarbashi, Hamid Reza. “*Design and Parametric Control of Co-Axes Driven Two-Wheeled Balancing Robot*”, Final Year Thesis, Massey University School of Engineering and Advanced Technology Albany New Zealand, 2010.
- [4] <http://mathworld.wolfram.com/TaylorSeries.html>
- [5] Rahmawatiy, Made., “ Stabilisasi Robot Pendulum Terbalik Beroda Dua Menggunakan Kontrol Fuzzyhybrid “, Tugas Akhir Teknik Elektro, Institut Teknologi Sepuluh Nopember, juni, 2014.
- [6] Martania A. D., “Implementasi Nonlinear Quadratic Tracking dengan Regresi Kuadratik untuk Inverted Pendulum,” Tugas Akhir. Teknik Elektro ITS, Surabaya, 2007.
- [7] Baik, Kyungjae. “*BoRam: Balancing Robot using Arduino and Lego*”, 2008

Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Nama : Farid Zainul Hasan
TTL : Situbondo, 08 Juni 1991
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat Rumah: Jl. Basuki Rahmat no. 257
Situbondo
Telp/HP : 085257277101
E-mail : *farid_smaji2006@yahoo.com*
farid14@gmail.ee.its.ac.id

RIWAYAT PENDIDIKAN

- 1997 – 2003 : SD Muhammadiyah 1 Situbondo
- 2003 – 2006 : SMP Muhammadiyah 1 Situbondo
- 2006 – 2009 : SMAN 1 Panji, Situbondo